IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yuheng Ju
2023.10

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection using web scraping and SpaceX API
  - Exploratory Data Analysis (EDA), including data wrangling, data visualization and interactive visual analytics
  - Machine learning prediction
- Summary of all results
  - The valuable data is based on the public sources
  - Exploratory Data Analysis (EDA) allowed to identify which features are the best to predict success of launchings
  - Machine Learning Prediction showed the best model to predict which characteristics are important to drive this opportunity by the best way, using all collected data.

Introduction

- Project Background:
- The commercial space age is here, companies are making space travel affordable for everyone. Space Y that would like to compete with SpaceX founded by Billionaire industrialist Allon Musk. The project is aimed to evaluate the viability of the new company Space Y to compete with Space X.
- Problems:
- Determining the price of each launch.
- Determining if SpaceX will reuse the first stage
- Determining if the first stage will land successfully
- Determining where to make the lauches

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data from Space X was obtained from 2 sources:
    - Space X API
      https://api.spacexdata.com/v4/rockets/
    - Web Scraping
      https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Perform data wrangling
  - Collected data was enriched by creating a landing outcome label based on outcome data after summarizing and analyzing features
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Scatter and bar graphs to show patterns between data

Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Data that was collected until this step were normalized, divided in training and test data sets and evaluated by four different classification models, being the accuracy of each model evaluated using different combinations of parameters.

Data Collection

- Data Collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes.

- Data from Space X was obtained from 2 sources:
  - Space X API
  https://api.spacexdata.com/v4/rockets/
  - Web Scraping
  https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Data Collection – SpaceX API

GitHub URL:
https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/1.%20data_%20collection.ipynb

Getting Data From API or Web Page → Making Data Frame From it → Filtering Data Frame as per requirement → Exporting to CSV

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	
89	86	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032
90	87	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032
91	88	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032
92	89	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033
93	90	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032

90 rows × 17 columns

Data Collection – SpaceX API

- Getting response from API

```
spacex_url = https://api.spacexdata.com/v4/launches/past
response = requests.get(spacex_url)
```

- Converting response to a json file

```
static_json_url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321ENSkillsNetwork/datasets/API/datasets/API_call_spacex_api.json’

response = requests.get(static_json_url)
```

- Applying custom function to clean data

Data Collection – SpaceX API

- Assigning list 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}
```

- Creating dataframe

```
data2 = pd.DataFrame(launch_dict)
```

- Filtering dataframe

```
data_falcon9 = data2[data2['BoosterVersion']!='Falcon 1’]
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
```

- Data Wrangling: using mean to replace null

```
data_falcon9['PayloadMass'].mean
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, 
payload_mean)
```

Data Collection – Web Scraping

GitHub URL:
https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/2.%20data_collection%20with_web_scraping.ipynb

1. Request the Falcon9 Launch Wiki page
2. Extract all column/variable names from the HTML table header
3. Create a data frame by parsing the launch HTML tables

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10
...	...	...	...	...	...	...	...	...	...	...	...
116	117	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1051.10	Success	9 May 2021	06:42
117	118	KSC	Starlink	~14,000 kg	LEO	SpaceX	Success\n	F9 B5B1058.8	Success	15 May 2021	22:56
118	119	File display CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success\n	F9 B5B1063.2	Success	26 May 2021	18:59
119	120	KSC	SpaceX CRS-22	3,328 kg	LEO	NASA	Success\n	F9 B5B1067.1	Success	3 June 2021	17:29
120	121	CCSFS	SXM-8	7,000 kg	GTO	Sirius XM	Success\n	F9 B5	Success	6 June 2021	04:26

Data Collection – SpaceX API

- Getting response from HTML

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

response = requests.get(static_url)
```

- Creating BeautifulSoup Object

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

- Finding tables

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

- Getting Column names

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

Data Collection – SpaceX API

- Appending data to the dictionary

```
launch_dict['Flight No.'].append(flight_number)
```

- Converting dictionary to dataframe

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

- Dataframe to CSV

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

Data Wrangling

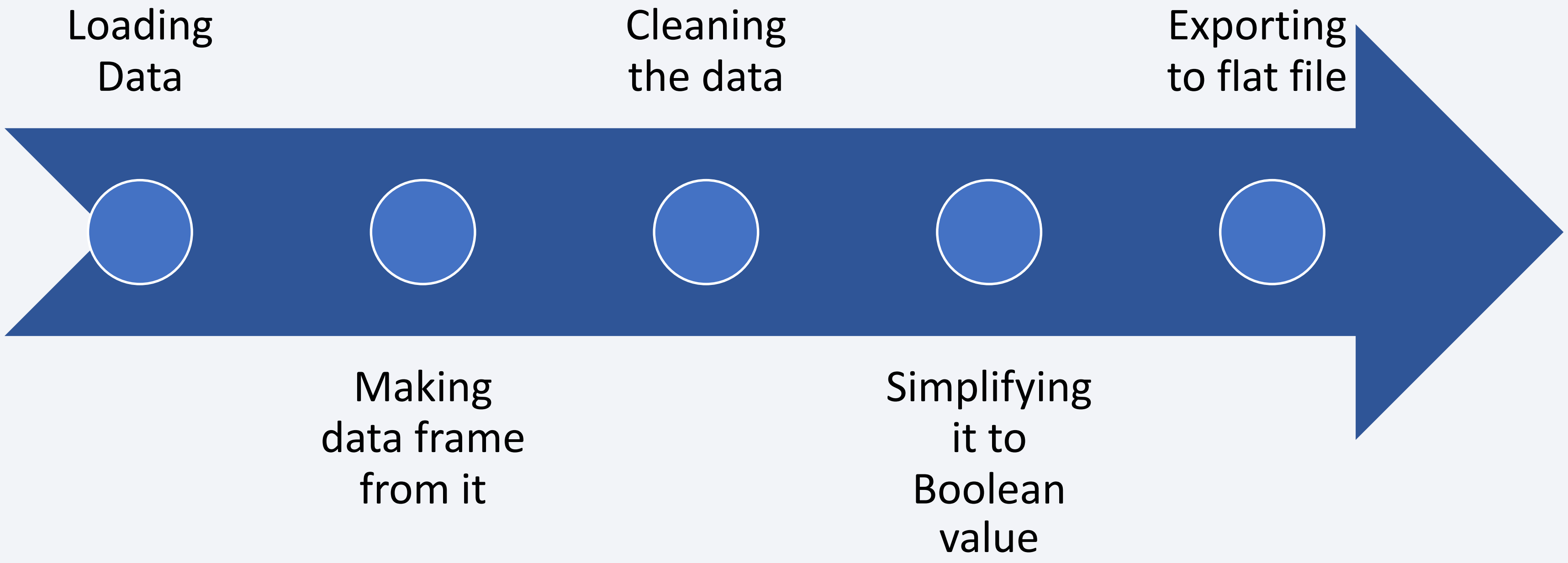
- Data Collection is the process of cleaning and unifying messy and complex data sets for easy access and analysis.

1. Calculate the number of launches on each site
   ```
   df['LaunchSite'].value_counts()
   ```
2. Calculate the number and occurrence of each orbit
   ```
   df['Orbit'].value_counts()
   ```
3. Calculate the number and occurrence of mission outcome of the orbits
   ```
   landing_outcomes = df['Outcome'].value_counts()
   df['Outcome'].value_counts()
   ```
4. Create a landing outcome label from Outcome column

```
landing_class = df['Outcome'].map(lambda x: 0 if x in bad_outcomes else 1)
df['Class']=landing_class
```

Data Wrangling

GitHub URL:
https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/3.%20data_wrangling.ipynb

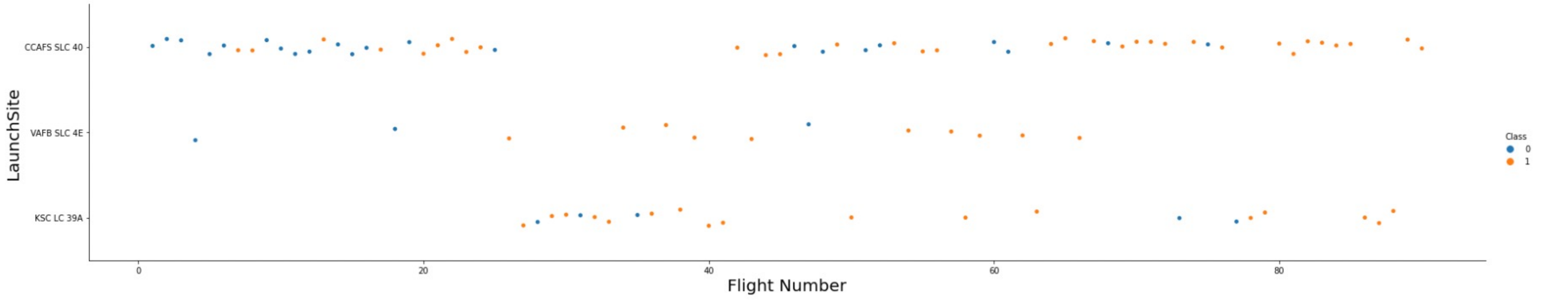


EDA with Data Visualization

GitHub URL:
https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/5.%20data_visualization.ipynb

Visualize the relationship between Flight Number and 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
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()
```

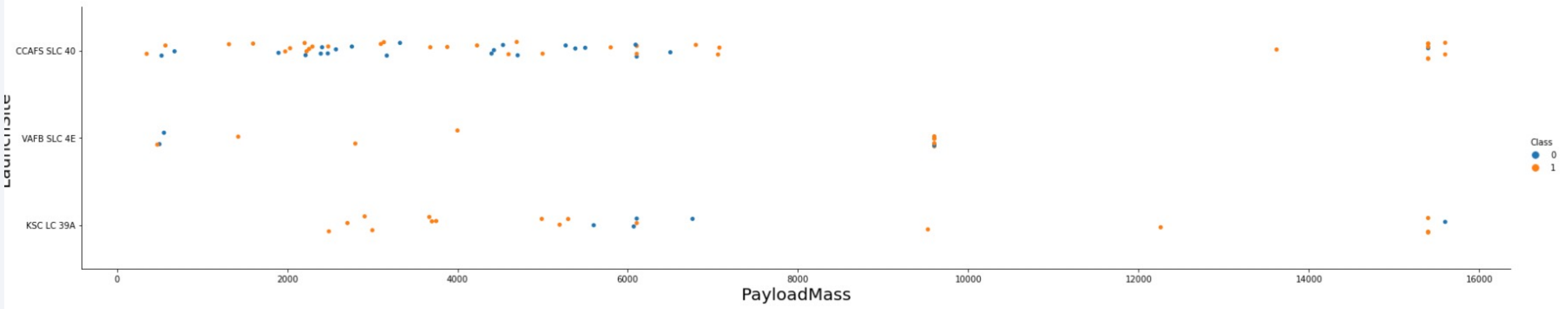


EDA with Data Visualization

GitHub URL:
https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/5.%20data_visualization.ipynb

Visualize the relationship between Payload and 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()
```



Now try to explain any patterns you found in the Payload Vs. Launch Site scatter point chart.

Most launches with payload mass under 10,000 kg are from any launch site, but heavier ones happens mainly at CCAFS SLC 40 and KSC LC 39A

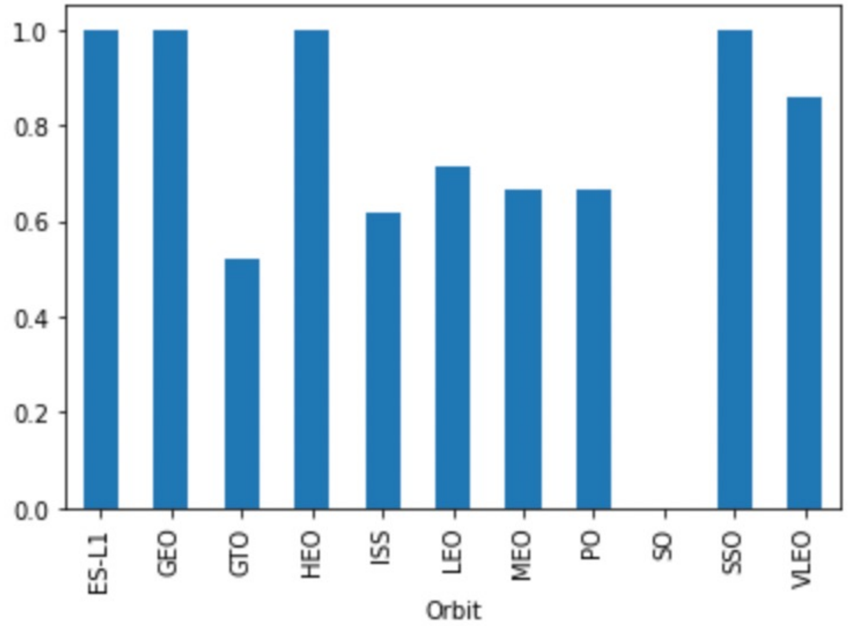
EDA with Data Visualization

GitHub URL:
https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/5.%20data_visualization.ipynb

Visualize the relationship between success rate of each orbit type

```
df.groupby('Orbit')['Class'].mean().plot.bar()
```

```
]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2169a52ad0>
```

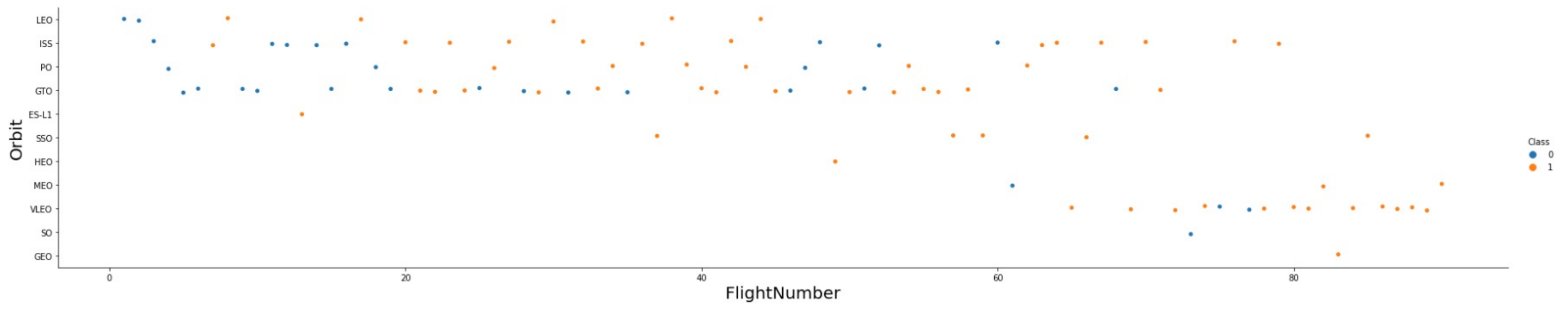


EDA with Data Visualization

GitHub URL:
https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/5.%20data_visualization.ipynb

Visualize the relationship between FlightNumber and Orbit type

```
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

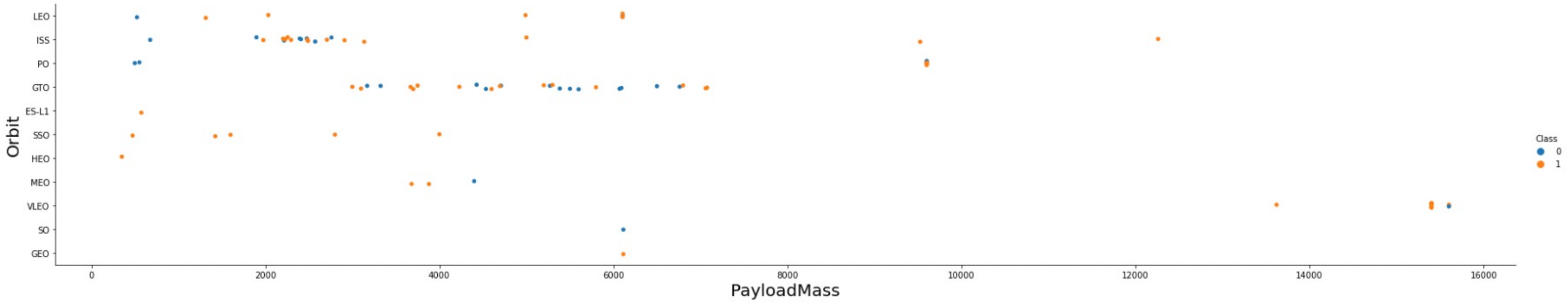


EDA with Data Visualization

GitHub URL:
https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/5.%20data_visualization.ipynb

Visualize the relationship between Payload and Orbit type

```python
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



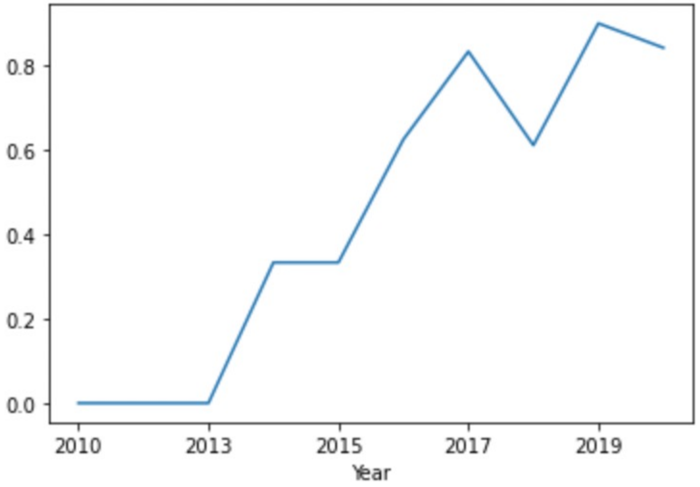
EDA with Data Visualization

GitHub URL:
https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/5.%20data_visualization.ipynb

Visualize the launch success yearly trend

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

```
[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2168e1c410>
```



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

EDA with SQL

- The following SQL queries were performed:
  - Names of the unique launch sites in the space mission;
  - Top 5 launch sites whose name begin with the string 'CCA';
  - Total payload mass carried by boosters launched by NASA (CRS);
  - Average payload mass carried by booster version F9 v1.1;
  - Datewhenthefirstsuccessfullandingoutcomeingroundpadwasachieved;
  - Names of the boosters which have success in drone ship and have payload mass between 4000 and 6000 kg;
  - Total number of successful and failure mission outcomes;
  - Names of the booster versions which have carried the maximum payload mass;
  - Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015; and
  - Rank of the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20.
- GitHub URL:
- https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/4.%20exploring_data.ipynb

Build an Interactive Map with Folium

- Folium makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map. We use the latitude and longitude coordinates for-each launch site and added a Circle Marker around each launch site with a label of the name of the launch site. It is also easy to visualize the number of success and failure for each launch site with Green and Red markers on the map.

GitHub URL:
https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/6.%20interactive_visual_with_Folium.ipynb

Build an Interactive Map with Folium

- Markers indicate points like launch sites;
- Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center;
- Marker clusters indicates groups of events in each coordinate, like launches in a launch site; and
- Lines are used to indicate distances between two coordinates.

Map Marker	Icon Maker	Circle Maker	PolyLine	Marker Cluster Object	AntPath
• folium.Marker() • Map objects to make a mark on the map	• folium.Icon() • Create an icon on the map	• folium.Circle() • Create a circle on the map	• folium.PolyLine() • Create a line between two points	• folium.Cluster() • Simplify a map containing many markers have the same coordinate	• folium.plugins.AntPath() • Create an animated line between two points

Build a Dashboard with Plotly Dash

- The following graphs and plots were used to visualize data
  - Percentage of launches by site
  - Payload range
- This combination allowed to quickly analyze the relation between payloads and launch sites, helping to identify where is best place to launch according to payloads.

- GitHub URL:
- https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/7.%20spacex_dash_app.py

Predictive Analysis (Classification)

- Four classification models were compared: logistic regression, support vector machine, decision tree and k nearest neighbors.

1. Building Models
    - Load our feature engineered data into dataframe
    - Transform it into NumPy arrays
    - Standardize and transform data
    - Split data into training and test data sets
    - Check how many test samples has been created
    - List down machine learning algorithms we want to use
    - Set our parameters and algorithms to GridSearchCV
    - Fit our datasets into the GridSearchCV objects and train our model

- GitHub URL:
- https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/8.%20machine_learning_prediction.ipynb

Predictive Analysis (Classification)

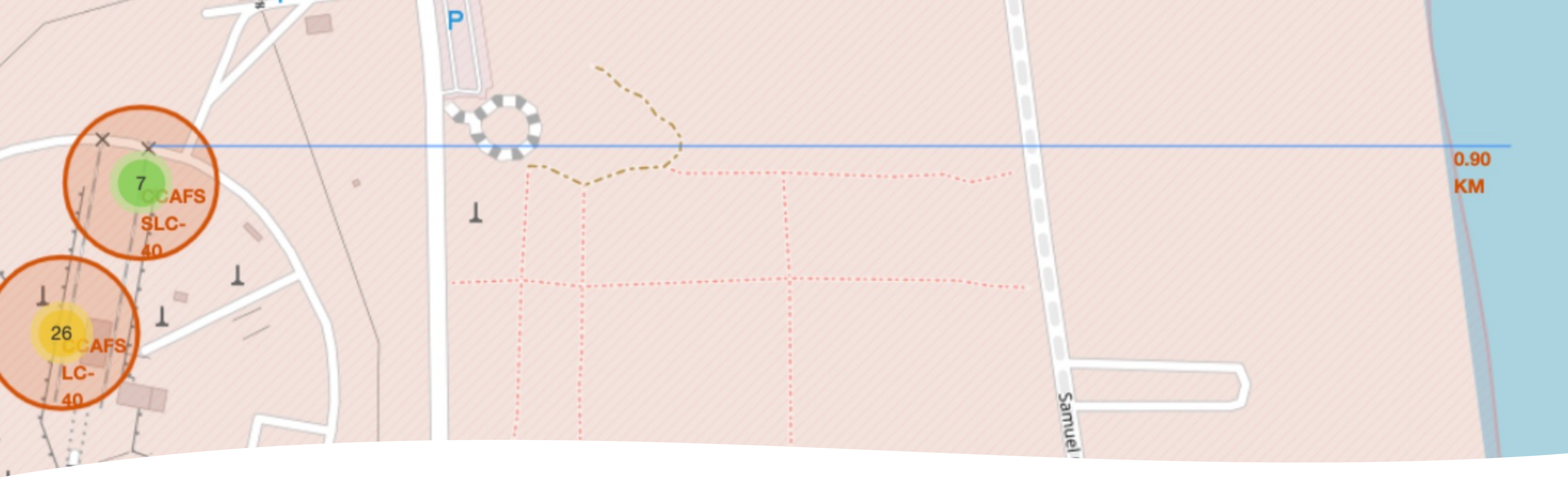
- Four classification models were compared: logistic regression, support vector machine, decision tree and k nearest neighbors.

2. Evaluating Models
    - Check accuracy for each model
    - Get best hyperppramaters for each type of algorithms
    - Plot Confusion Matrix
3. Finding Best performing Classification Models
    - The model with the best accuracy score wins the best performing model
4. Best Model

- GitHub URL:
- https://github.com/YuhengJu/coursera_IBM_DS_Capstone_SpaceY/blob/main/8.%20machine_learning_prediction.ipynb

Results

- Exploratory data analysis results:
  - Space X uses 4 different launch sites;
  - The first launches were done to Space X itself and NASA;
  - The average payload of F9 v1.1 booster is 2,928 kg;
  - The first success landing outcome happened in 2015 fiver year after the first launch;
  - Many Falcon 9 booster versions were successful at landing in drone ships having payload above the average;
  - Almost 100% of mission outcomes were successful;
  - Two booster versions failed at landing in drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015;
  - The number of landing outcomes became as better as years passed.

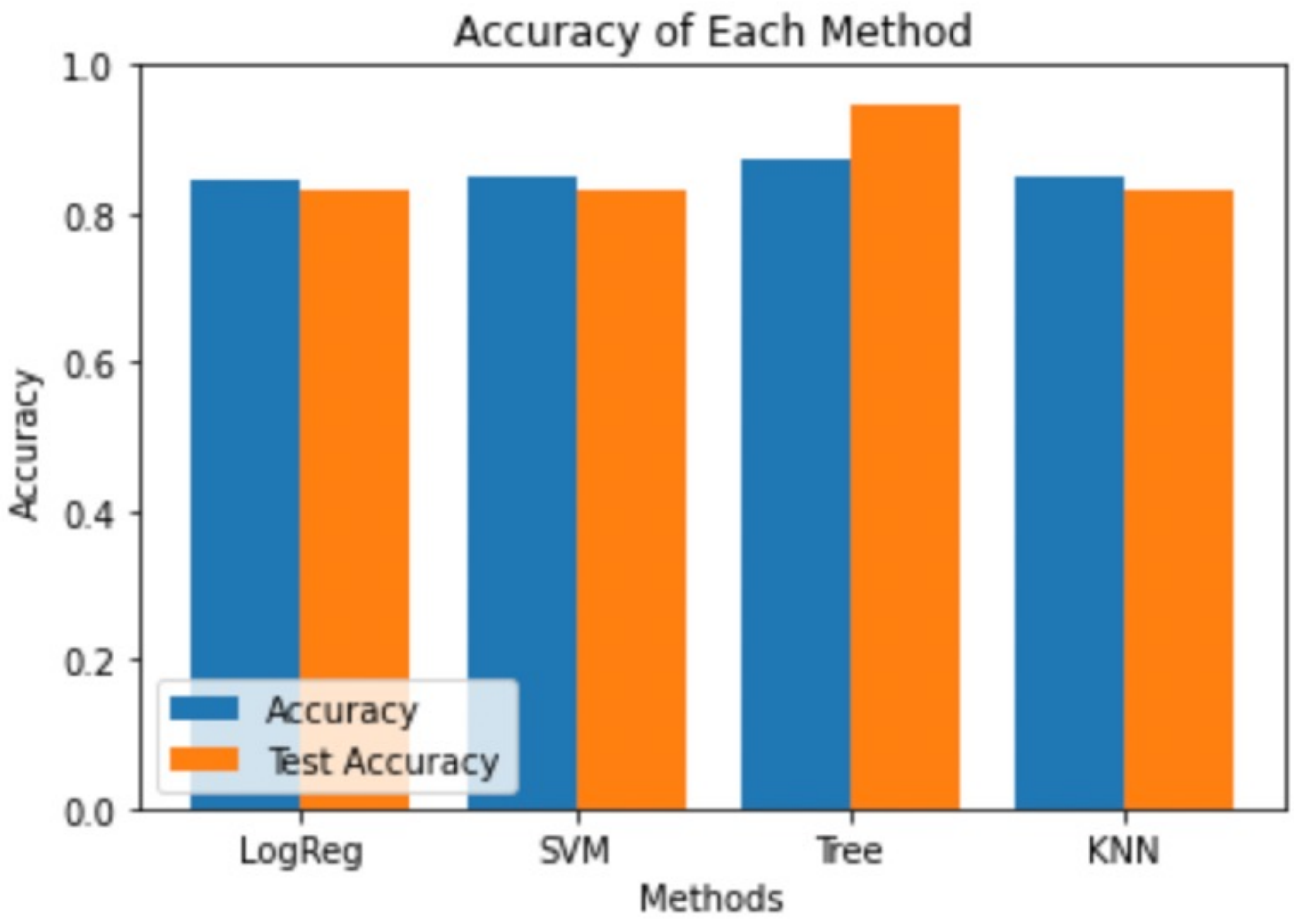


Results

- Interactive Analytics demo in screen shoots:
  - Using interactive analytics was possible to identify that launch sites use to be in safety places, near sea, for example and have a good logistic infrastructure around.
  - Most launches happens at east cost launch sites.

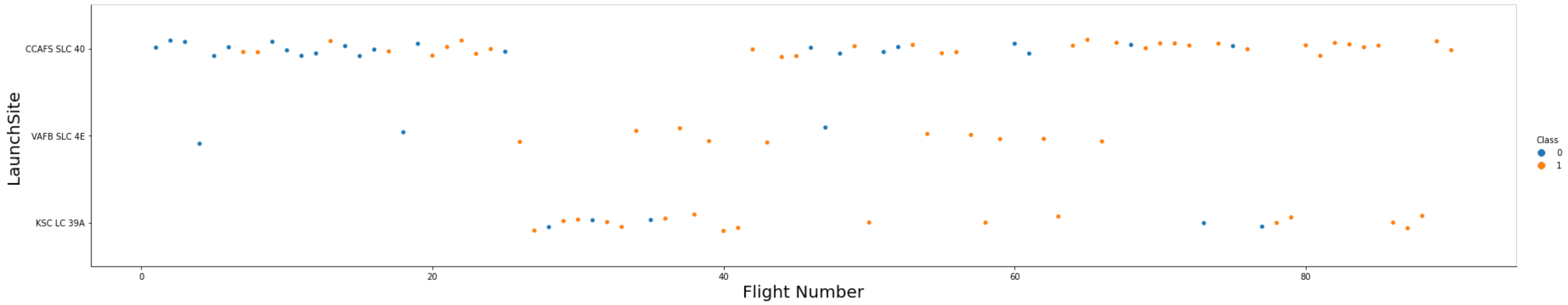
Results

- Predictive Analysis results:
  - Predictive Analysis showed that Decision Tree Classifier is the best model to predict successful landings, having accuracy over 87% and accuracy for test data over 94%.



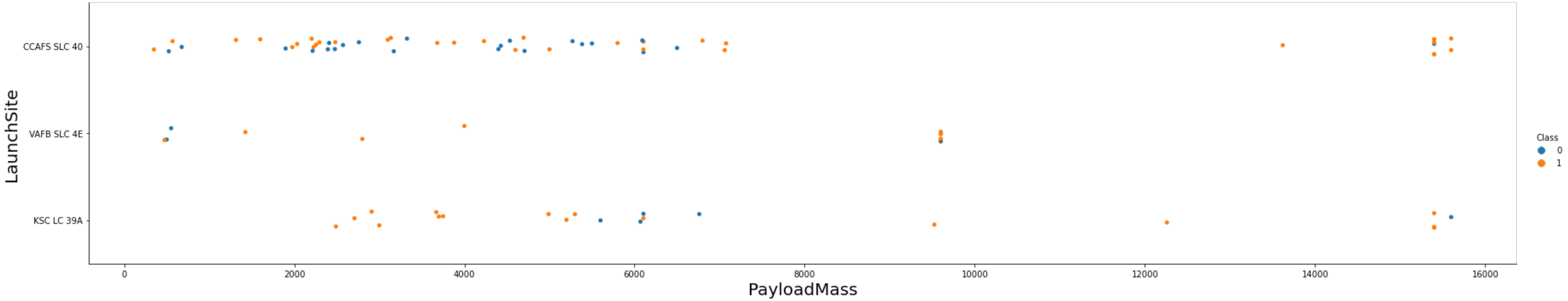
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

- According to the plot above, it's possible to verify that the best launch site nowadays is CCAF5 SLC 40, where most of recent launches were successful;
- In second place VAFB SLC 4E and third place KSC LC 39A;
- It's also possible to see that the general success rate improved over time.
- when flight number increases, the success rate for rockets increases.

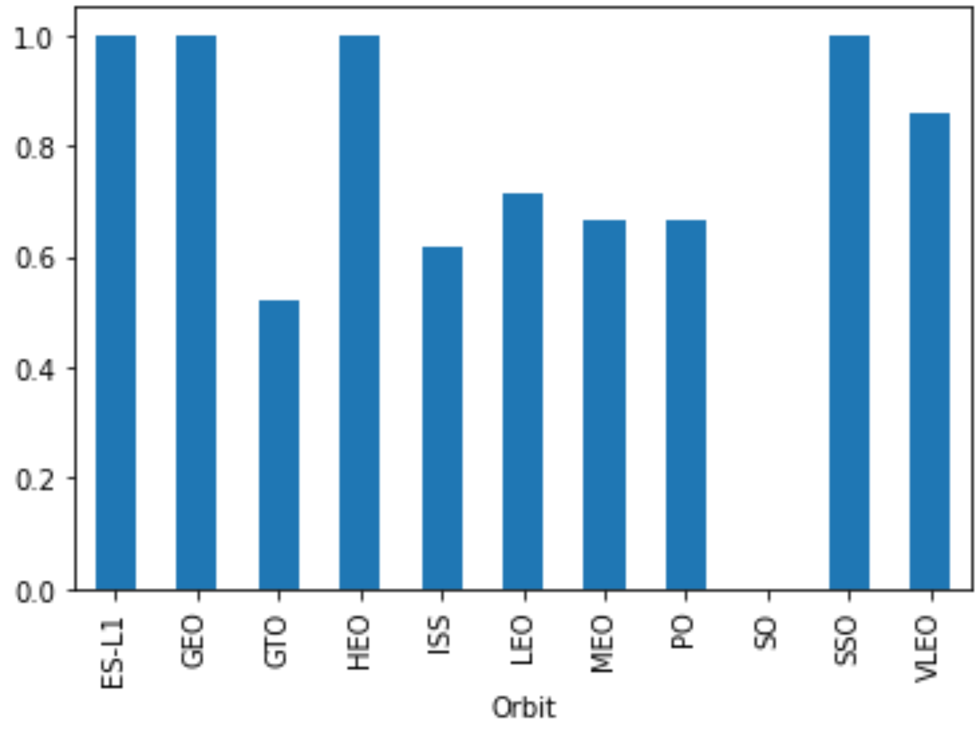


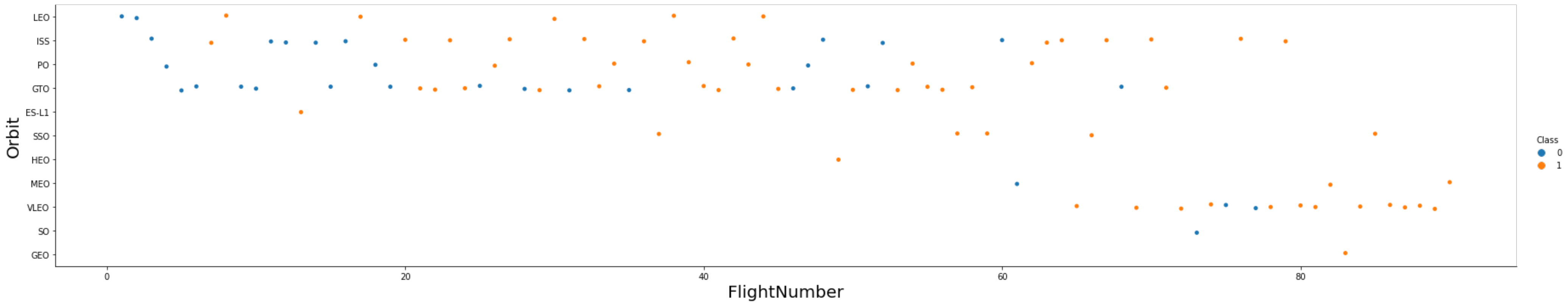
Payload vs. Launch Site

- Payloads over 9,000kg (about the weight of a school bus) have excellent success rate;
- Payloads over 12,000kg seems to be possible only on CCAFS SLC 40 and KSC LC 39A launch sites.

Success Rate vs. Orbit Type

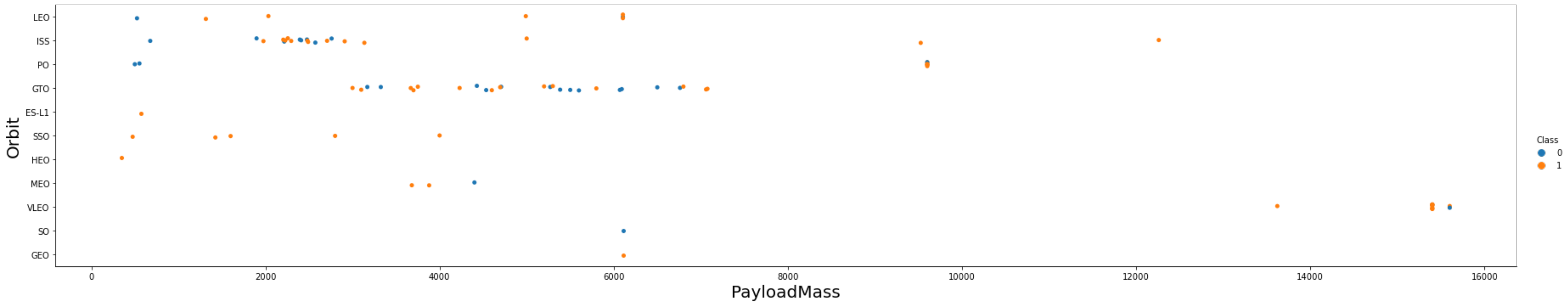
- The biggest success rates happens to orbits:
  - ES-L1;
  - GEO;
  - HEO;
  - SSO.
- Followed by:
  - VLEO (above 80%);
  - LFO(above70%).





Flight Number vs. Orbit Type

- Apparently, success rate improved over time to all orbits;
- VLEO orbit seems a new business opportunity, due to recent increase of its frequency.

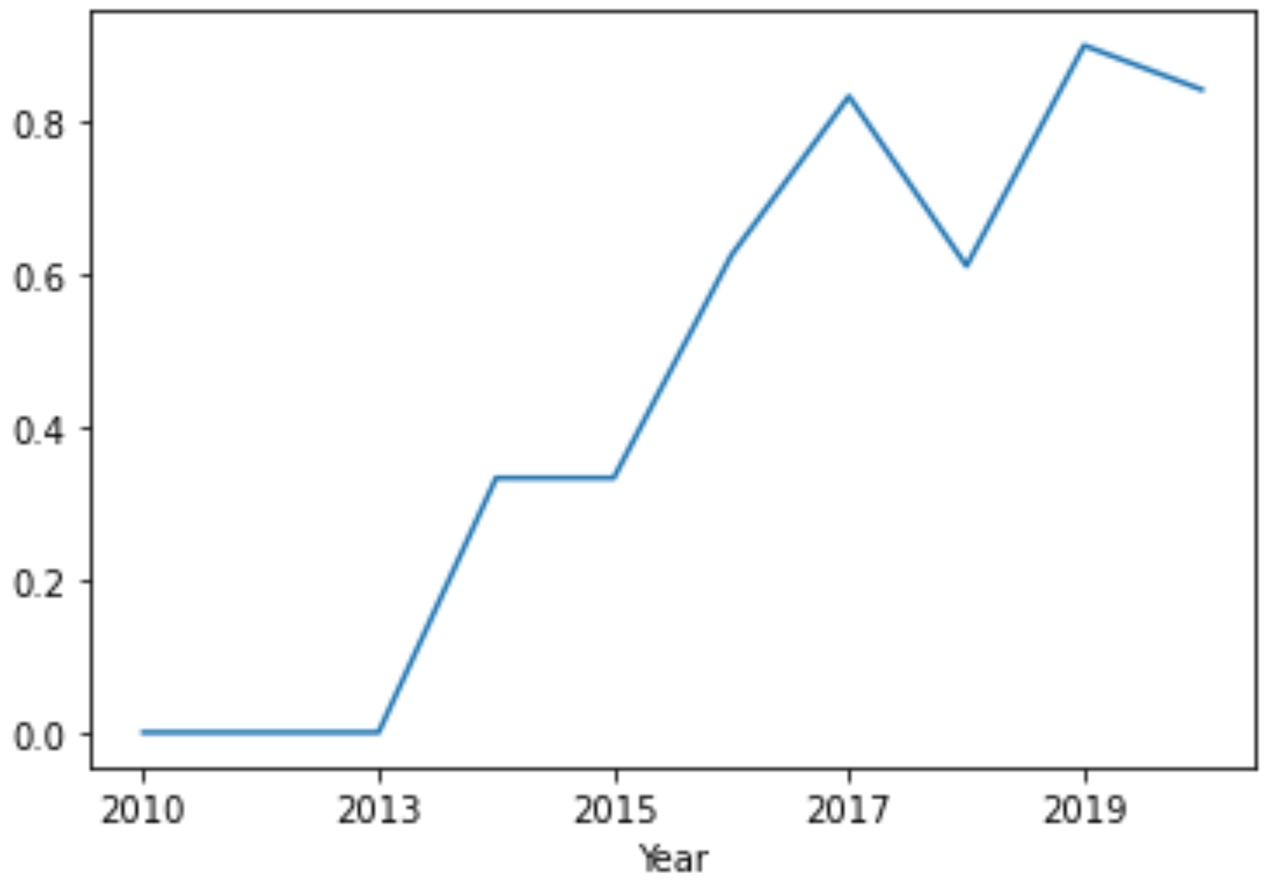


Payload vs. Orbit Type

- Apparently, there is no relation between payload and success rate to orbit GTO;
- ISS orbit has the widest range of payload and a good rate of success;
- There are few launches to the orbits SO and GEO.

Launch Success Yearly Trend

- Success rate started increasing in 2013 and kept until 2020;
- It seems that the first three years were a period of adjusts and improvement of technology.



All Launch Site Names

- According to data, there are four launch sites:
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E
- They are obtained by selecting unique occurrences of “launch site” values from the dataset.

```
sql SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL ORDER BY 1;
```

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`.
- Here we can see five samples of Cape Canaveral launches.
- Using keyword LIMIT 5 in query we fetch 5 records from table spaceX and with condition LIKE with wild card CCA%
- `sql 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__o
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (pa
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 (pa
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No

Total Payload Mass

- Total payload carried by boosters from NASA:
- Total payload calculated above, by summing all payloads whose codes contain ‘CRS’, which corresponds to NASA.

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

total_payload
111268

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1:
- Filtering data by the booster version above and calculating the average payload mass we obtained the value of 2,928 kg.
- `sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';`

avg_payload
2928

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad
- By filtering data by successful landing outcome on ground pad and getting the minimum value for date it’s possible to identify the first occurrence, that happened on 12/22/2015.
- `sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';`

first_success_gp
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Selecting distinct booster versions according to the filters above, these 4 are the result.
- `sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Success (drone ship)'`

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

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes
- Grouping mission outcomes and counting records for each group led us to the summary above.
- `sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;`

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

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass
- These are the boosters which have carried the maximum payload mass registered in the dataset.
- sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL) ORDER BY BOOSTER_VERSION;

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

Boosters Carried Maximum Payload

- The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND DATE_PART('YEAR', DATE) = 2015;

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

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

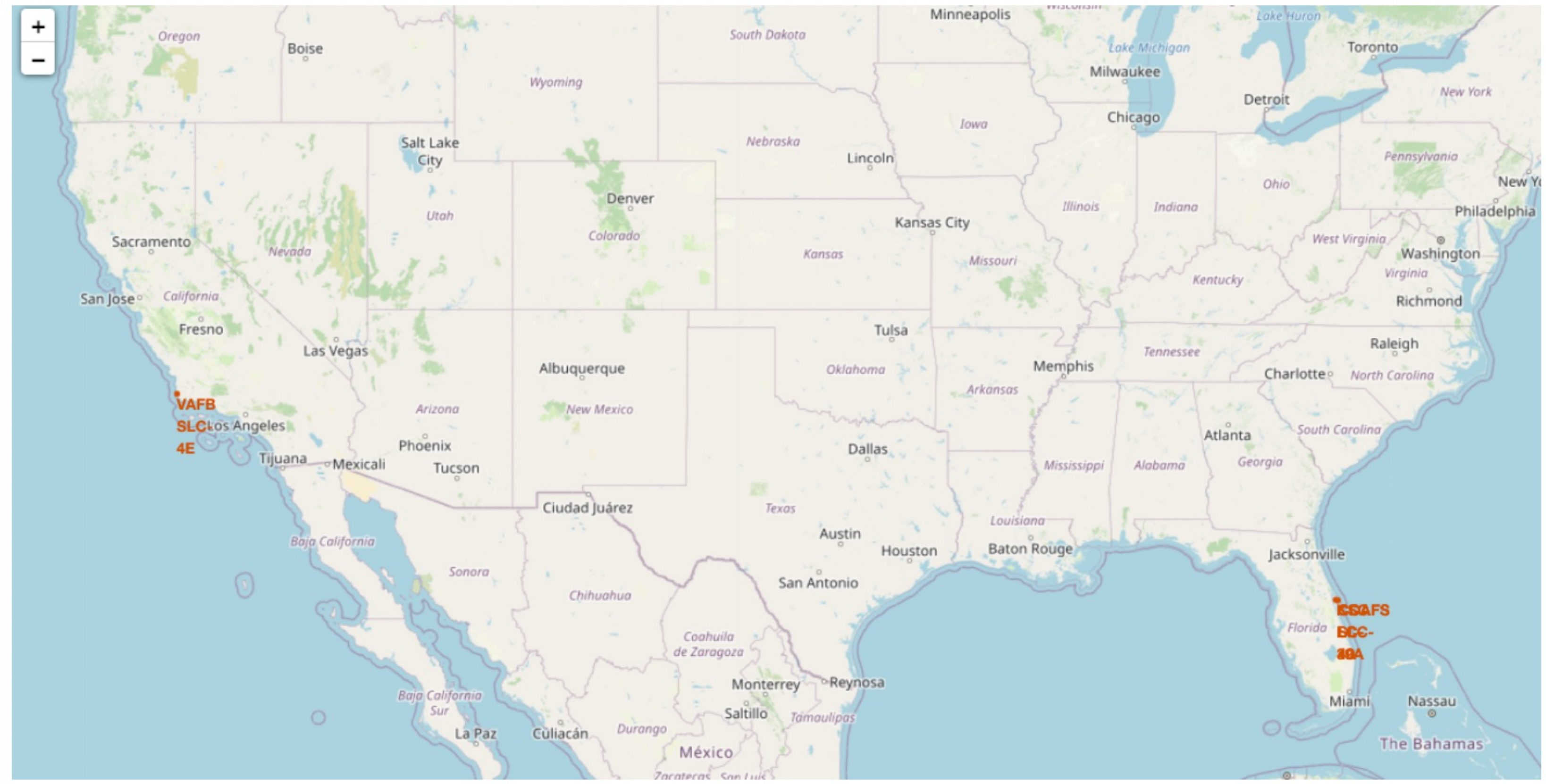
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- This view of data alerts us that "No attempt" must be taken in account.
- `sql SELECT LANDING__OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY QTY DESC;`

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

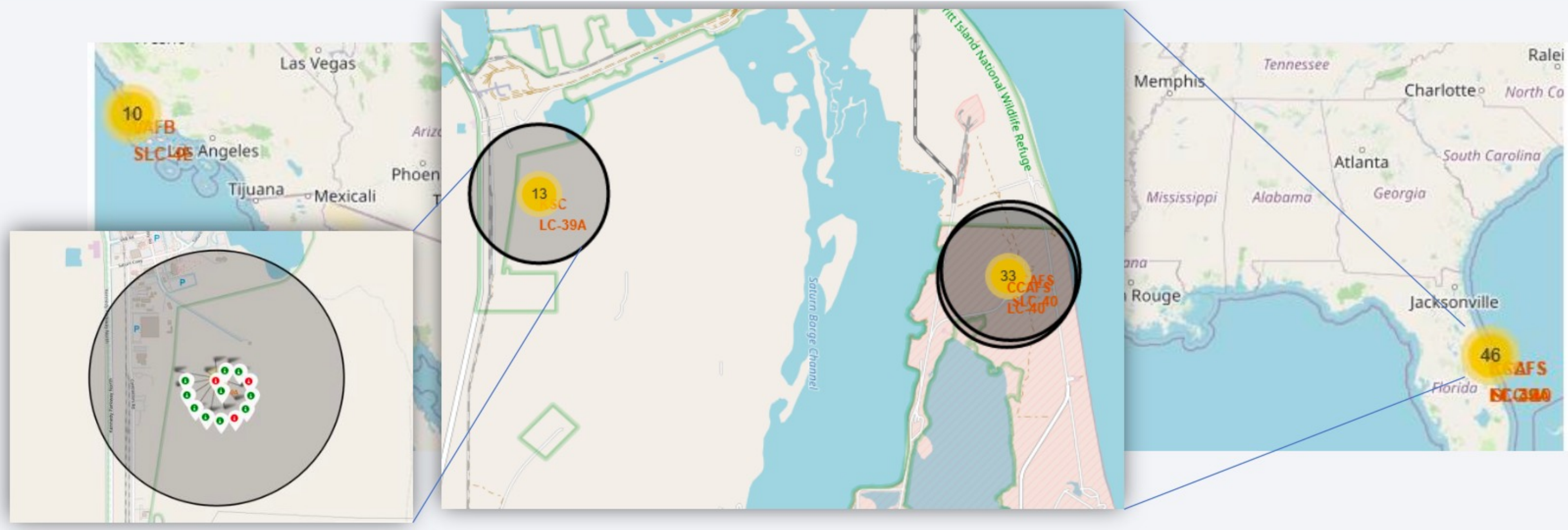
Launch Sites Proximities Analysis

All Launch Sites



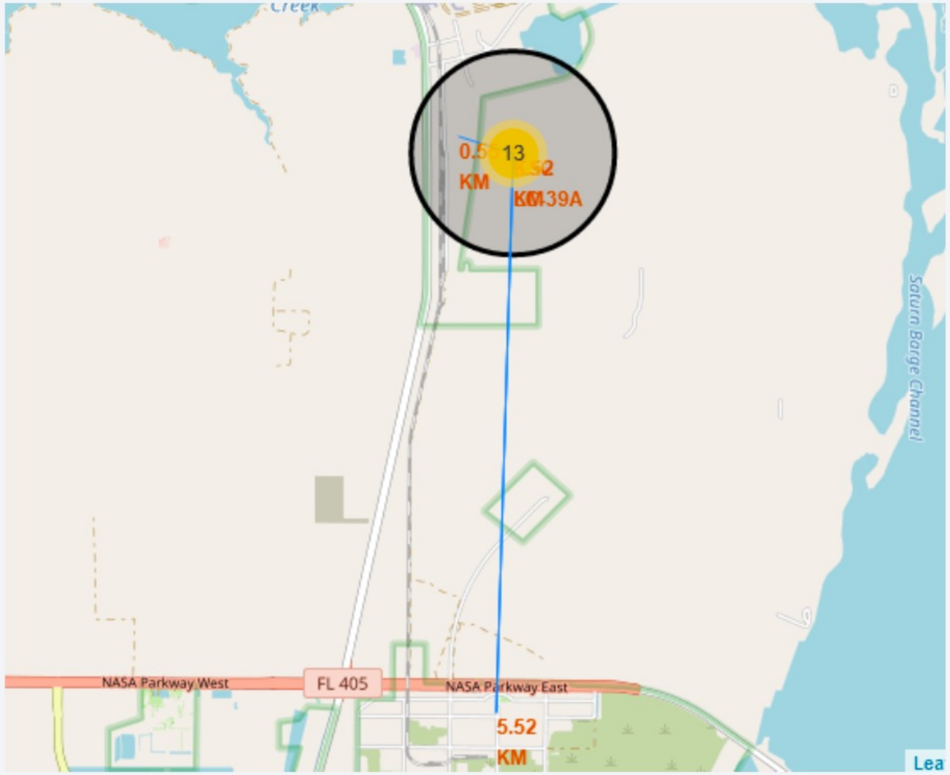
We can see that spaceX lauch sites are near to the US of coasts.

Launch Outcomes by Sites



Example of KSC LC-39A launch site launch outcomes. Green markers indicate successful and red ones indicate failure.

Logistics and Safety

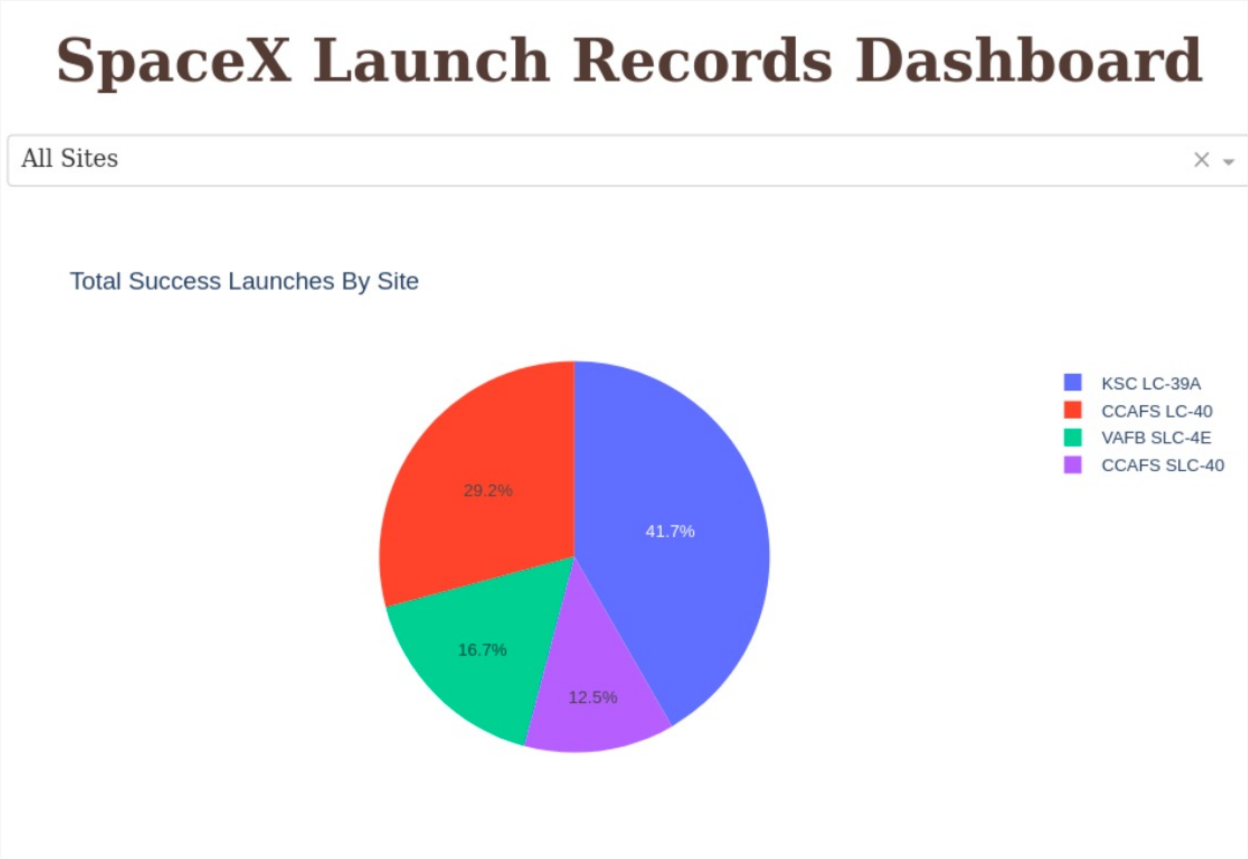


Launch site KSC LC-39A has good logistics aspects, being near railroad and road and relatively far from inhabited areas.

Section 4

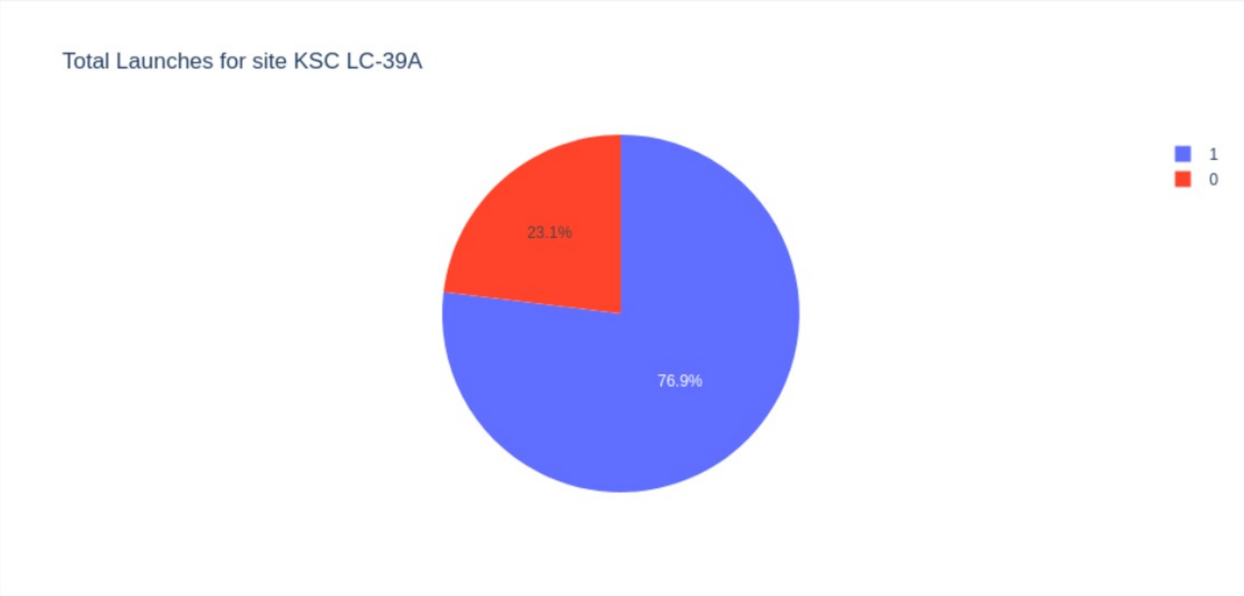
Build a Dashboard with Plotly Dash

Successful Launches by Site



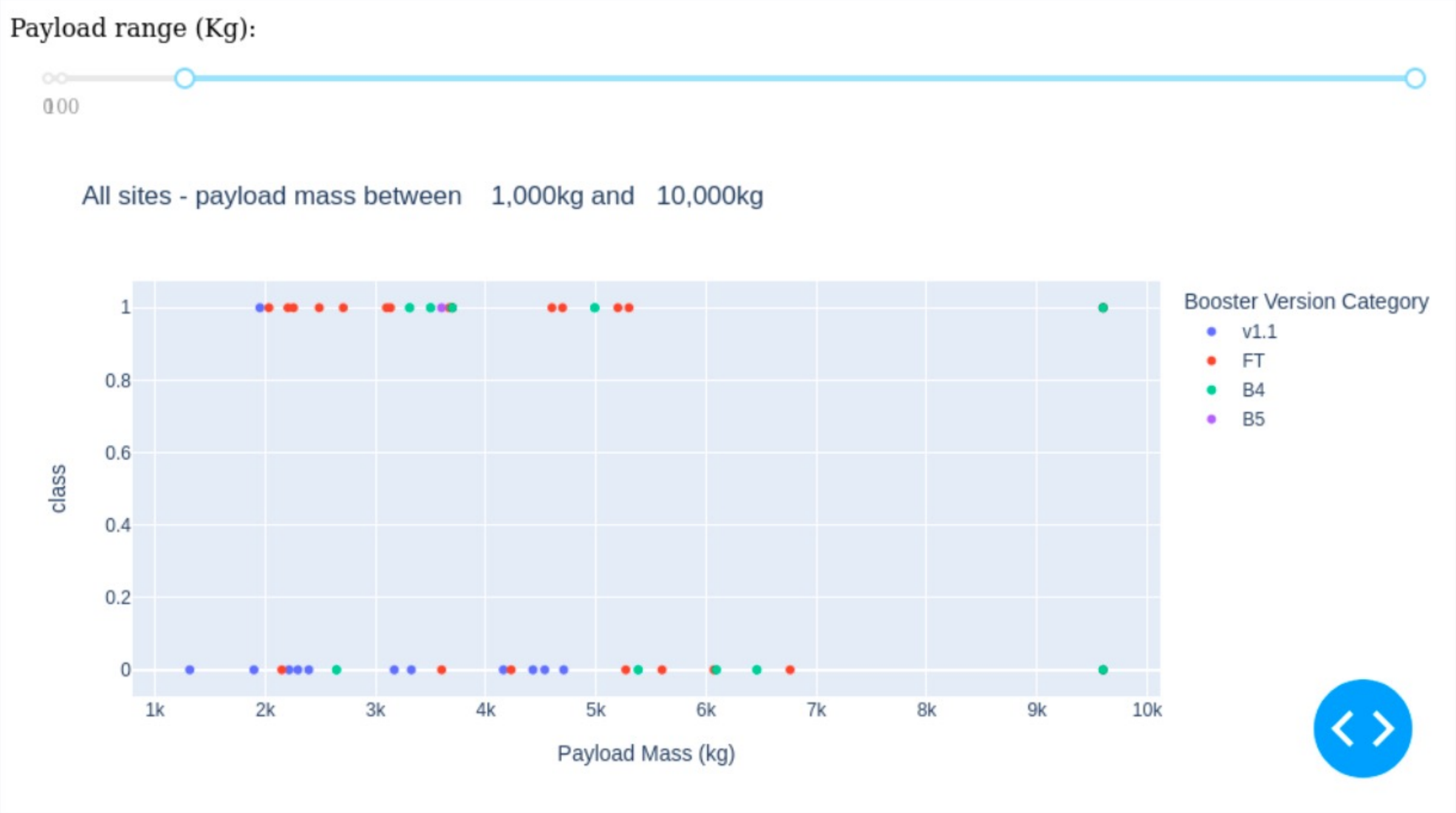
- The place from where launches are done seems to be a very important factor of success of missions.

Launch Success Ratio for KSC LC-39A



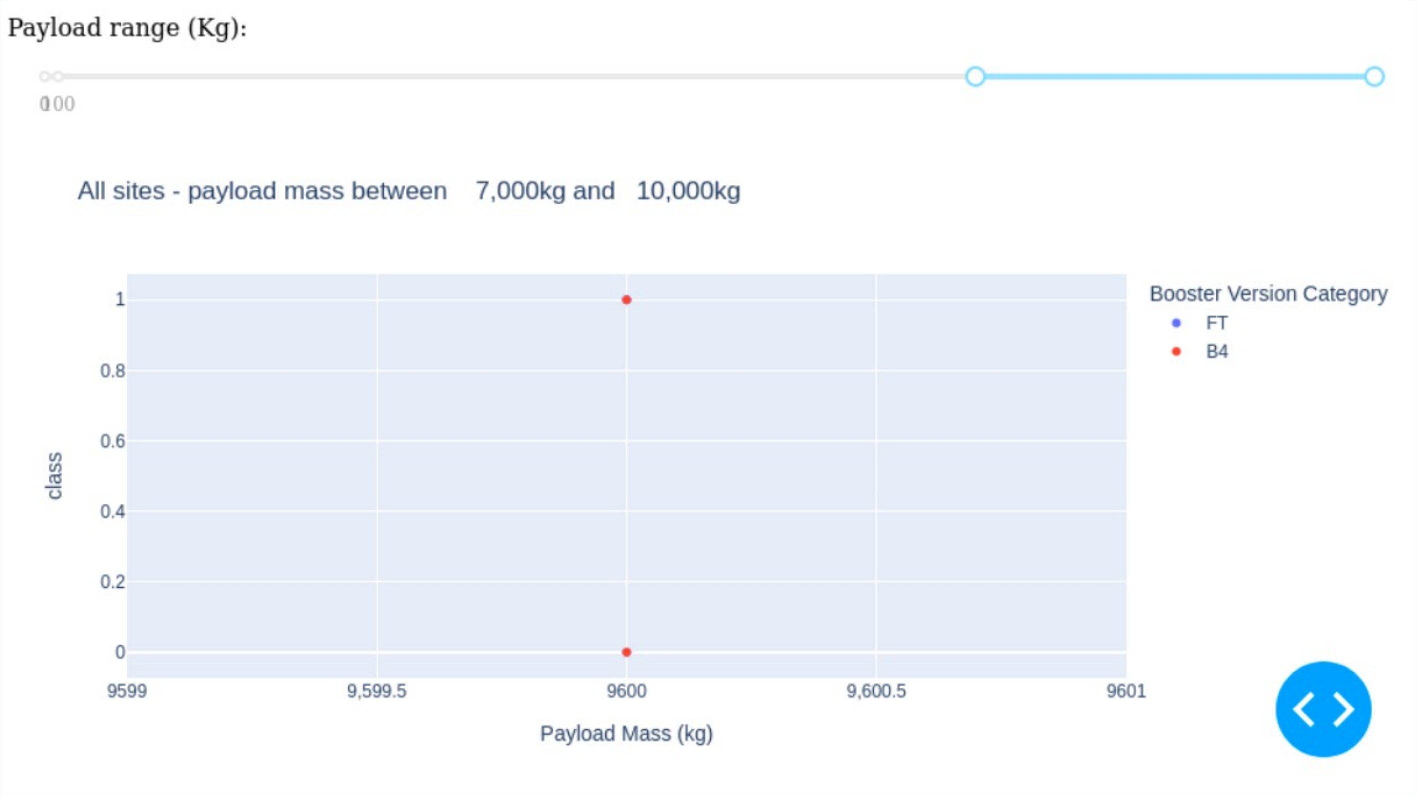
- 76.9% of launches are successful in this site.

Payload vs. Launch Outcome



- Payloads under 6,000kg and FT boosters are the most successful combination.

Payload vs. Launch Outcome



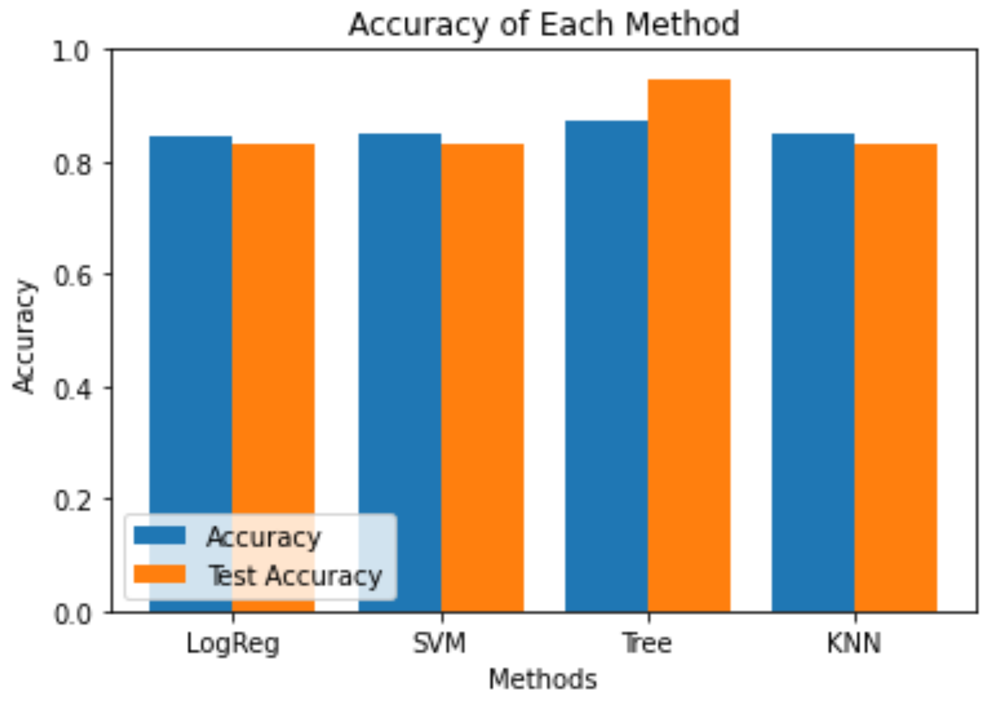
- There's not enough data to estimate risk of launches over 7,000kg

Section 5

Predictive Analysis (Classification)

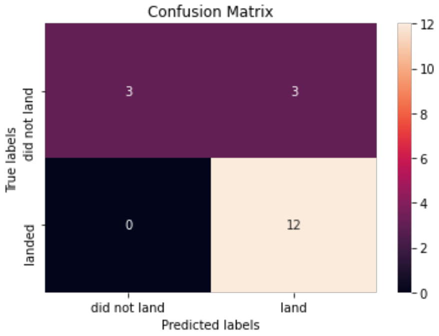
Classification Accuracy

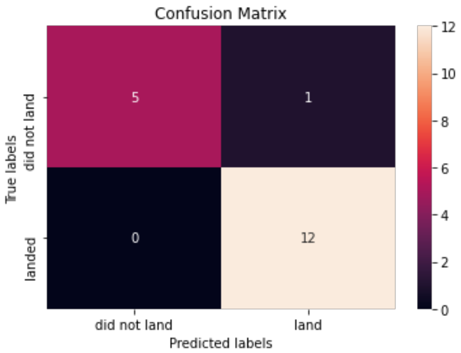
- Four classification models were tested, and their accuracies are plotted beside;
- The model with the highest classification accuracy is Decision Tree Classifier, which has accuracies over than 87%.

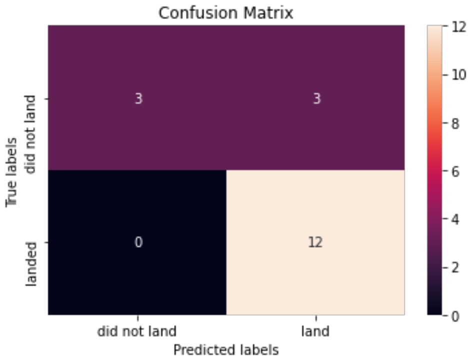


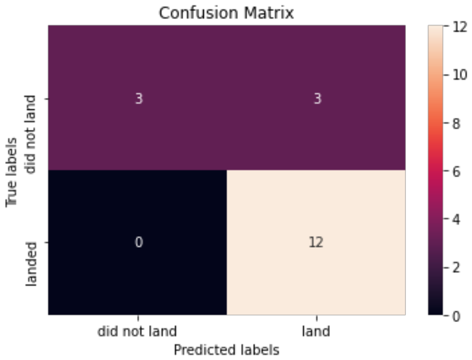
Confusion Matrix

- Confusion matrix of Decision Tree Classifier proves its accuracy by showing the big numbers of true positive and true negative compared to the false ones.









Conclusions

- Different data sources were analyzed, refining conclusions along the process;
- The best launch site is KSC LC-39A;
- Launches above 7,000kg are less risky;
- Although most of mission outcomes are successful, successful landing outcomes seem to improve over time, according the evolution of processes and rockets;
- Decision Tree Classifier can be used to predict successful landings and increase profits.

Thank you!