IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

CJ Zhao
Nov 23, 2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- We used a variety of methodologies:
  - Used API to request data, and also web scraping using BeautifulSoup
  - SQL Queries to do exploratory data analysis, data wrangling to handle missing value
  - Folium to visualize data in a more interactive way;
  - Use various Python packages such as pandas, seaborn & sklearn.
  - Standardize features prior to model fit, train_test_split to split data.
  - Machine Learning classification algorithms to predict if a land would be successful or not.
- Summary of all results :
  - SO & GTO have low success rates while orbit types like ESL1, GEO, HEO and SSO have 100% success rates
  - CCAFS SLC-40 has the highest success rate (at 42.9%) among all launch sites
  - On average, Booster Version ‘RT’ has highest success rate, while Booster Version V1.1 has lowest success rate.
  - And it is not the case that the higher the pay load, the higher the success rates.

Introduction

- SpaceX states that Falcon 9 rocket launches cost 62 million dollars each, while other providers cost upward of 165 million dollars each.
- Much of the savings for Falcon 9 rocket launches is because SpaceX can reuse the first stage. So if we can determine if the first stage would be successful, we can determine the cost of a launch.
- We collect historical data on SpaceX launches, and train machine learning models to predict the success rate of first-stage rocket landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Request using SpaceX public API
    - Web scraping using BeautifulSoup
- Perform data wrangling
    - Drop non-useful columns, replace missing values with the mean value of the features
    - Label the data: classify the landing outcome into success (1) and failure (0)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using ML classification models
    - Logistic Regression, SVM, Decision Tree, KNN Clusters
    - Fine-tuned the model parameters using GridSearchCV

Data Collection

- Data was collected mainly in two ways: API call and Web Scraping

Spacex API Request → Normalize JSON data → Create DataFrame → Pre-Processing Data

[CapStone-Project/spacex-data-collection-api.ipynb at main · caijiao314159/CapStone-Project (github.com)]()

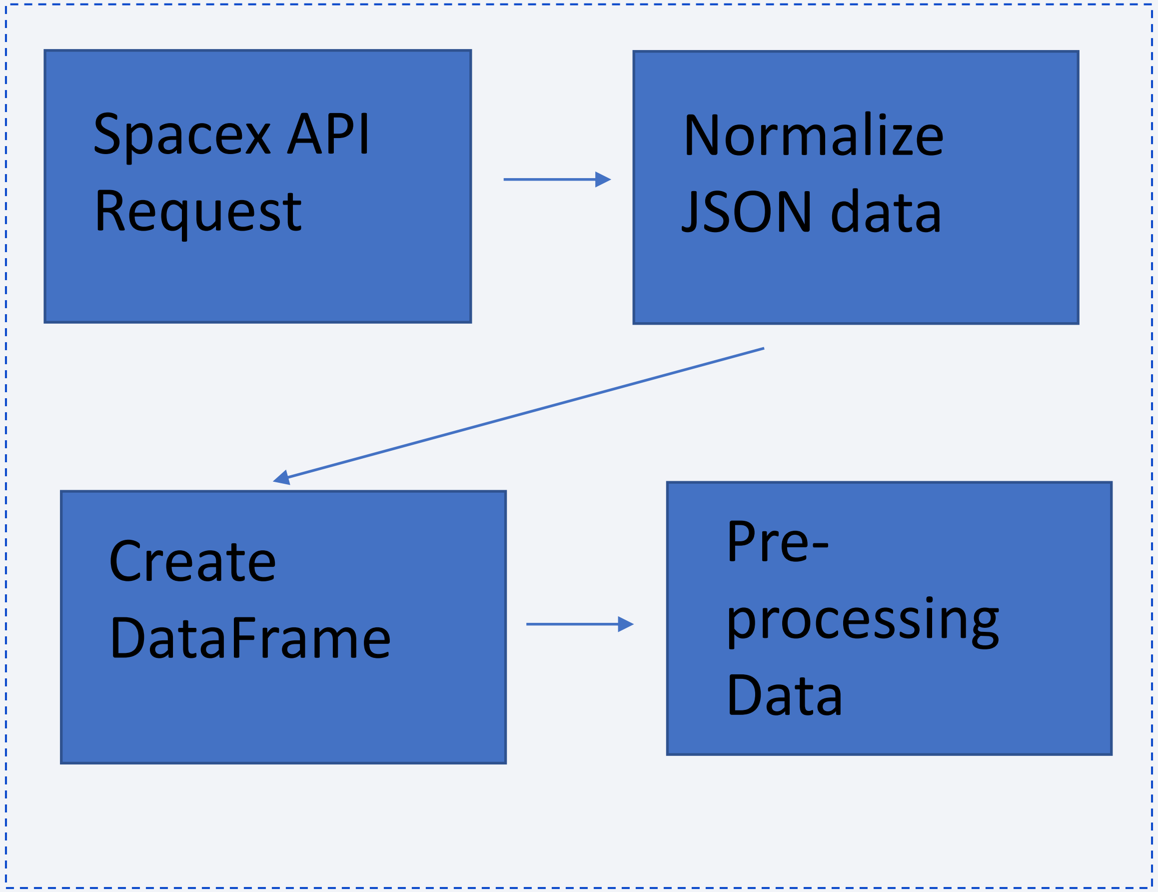
Request using URL → Create BeautifulSoup Object → Parse HTML Table → Create DataFrame

[CapStone-Project/spacex-data-collection-webscraping.ipynb at main · caijiao314159/CapStone-Project (github.com)]()

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
  - response = requests.get(spacex_url)
  - data=pd.json_normalize(response.json())
  - launch_data=pd.DataFrame(launch_dict)
  - Dealing with missing data: e.g., replace(np.nan, mean_PayLoanMass)

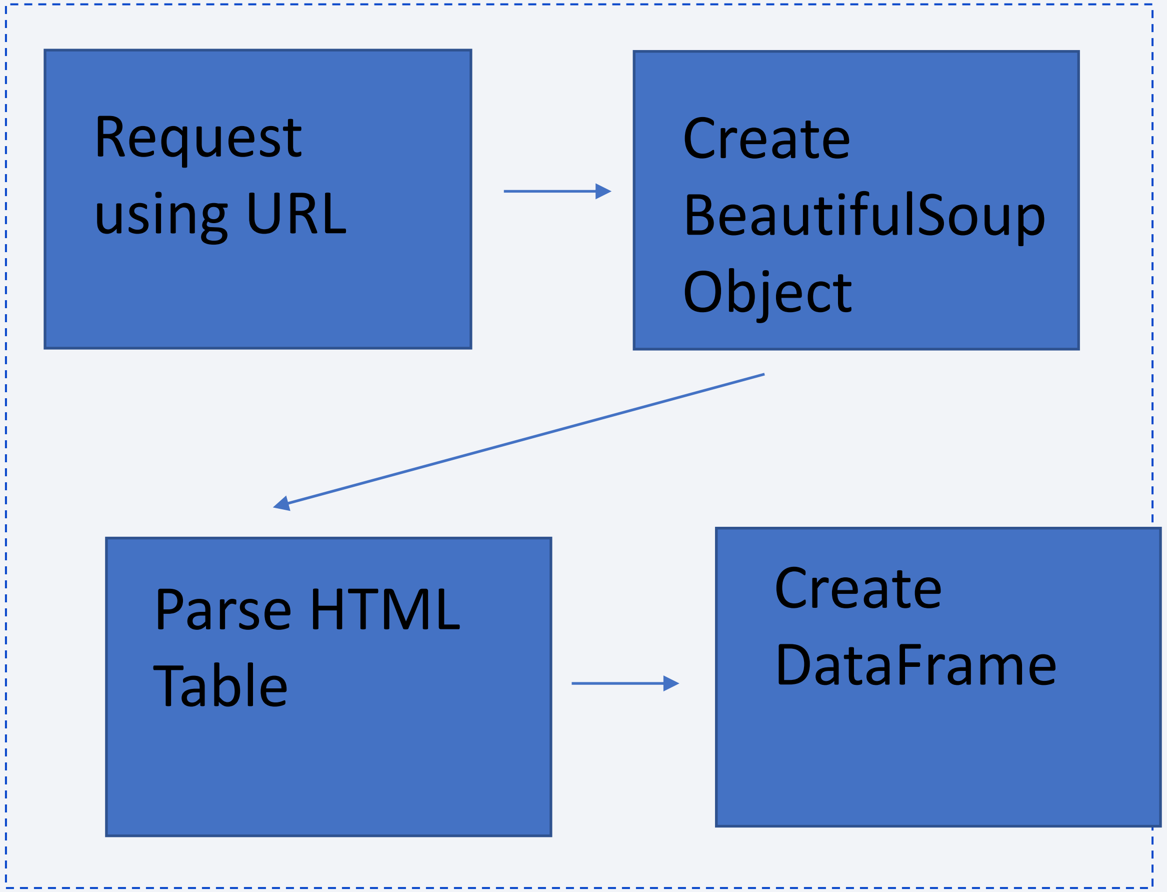
[CapStone-Project/spacex-data-collection-api.ipynb at main · caijiao314159/CapStone-Project (github.com)]()



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
  - page = requests.get(static_url).text
  - soup = BeautifulSoup(page,'html.parser’)
  - html_tables = soup.find_all("table")
  - table.find_all("tr"):
  - df=pd.DataFrame(launch_dict)

[CapStone-Project/spacex-data-collection-webscraping.ipynb at main · caijiao314159/CapStone-Project (github.com)]()



Data Wrangling

- Bring data into Numpy array, for simple exploratory data analysis (EDA)
  - resp = await fetch(URL)
  - dataset_part_1_csv = io.BytesIO((await resp.arrayBuffer()).to_py())
  - df=pd.read_csv(dataset_part_1_csv)
  - df.isnull().sum()/df.count()
  - df['LaunchSite'].value_counts()
- Label the outcome as 0 or 1
  - df['Class']=landing_class
  - df["Class"].mean()

Fetch CSV data into Numpy Array → Exploratory Data Analysis → Label the outcome → Calculate average success rate

- [CapStone-Project/spacex-data_wrangling.ipynb at main · caijiao314159/CapStone-Project (github.com)](https://github.com/caijiao314159/CapStone-Project)

EDA with Data Visualization

- EDA was performed on the variables Flight Number, Payload Mass, Launch Site, Orbit, Class, and Year.

- We did scatter plots on various features to assess their relationships:
  - FlightNumber vs. PayloadMass,
  - FlightNumber vs. LaunchSite,
  - LaunchSite vs. PayloadMass,
  - FlightNumber vs. Orbit,
  - PayloadMass vs. Orbit
- Class was the success classification, the 0 -1 label.
  - Bar Chart: Orbit vs. Succes
  - Line Chart: annual successrate

https://github.com/caijiao314159/CapStone-Project/blob/main/spacex-data-visualization.ipynb

EDA with SQL

- The SQL queries performed include:
  - SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
  - SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
  - SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)’;
  - SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version='F9 v1.1’;
  - SELECT MIN(Date) FROM SPACEXTBL WHERE [Landing _Outcome]= 'Success (ground pad)’;
  - SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE [Landing _Outcome]= 'Success (drone ship)' AND 4000 < PAYLOAD_MASS__KG_ < 6000;
  - SELECT CASE WHEN MISSION_OUTCOME LIKE 'Success%' then 'Success' else 'Failure' end as Outcome, count(MISSION_OUTCOME) as Total_number FROM SPACEXTB
  - SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
  - select substr(date,4,2) as month_Name, [Landing _Outcome], BOOSTER_VERSION, Launch_Site FROM SPACEXTBL WHERE [Landing _Outcome]='Failure (drone ship)' and substr(Date,7,4)='2015'
  - SELECT [Landing _Outcome], COUNT(case when MISSION_OUTCOME LIKE 'Success%' then 'Success' end) Total_number FROM SPACEXTBL GROUP BY [Landing _Outcome] ORDER BY TOTAL_NUMBER DESC

- https://github.com/caijiao314159/CapStone-Project/blob/main/eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We used Folium maps (longitude and latitude) to mark all the launch site.
  - circle = folium.Circle(nasa_coordinate, radius=1000, color='#d35400', fill=True).add_child(folium.Popup('NASA Johnson Space Center’))
  - site_map = folium.Map(location=nasa_coordinate, zoom_start=10)
  - site_map.add_child(circle)
  - site_map.add_child(marker)
- We computed distance between launch site and Central Command center
- We marked the successful and failed launchings for each site using color coding.
  - This would allow us to assess the impact of launch site on success rate

https://github.com/caijiao314159/CapStone-Project/blob/main/spacex-Folium-Analysis.ipynb

Build a Dashboard with Plotly Dash

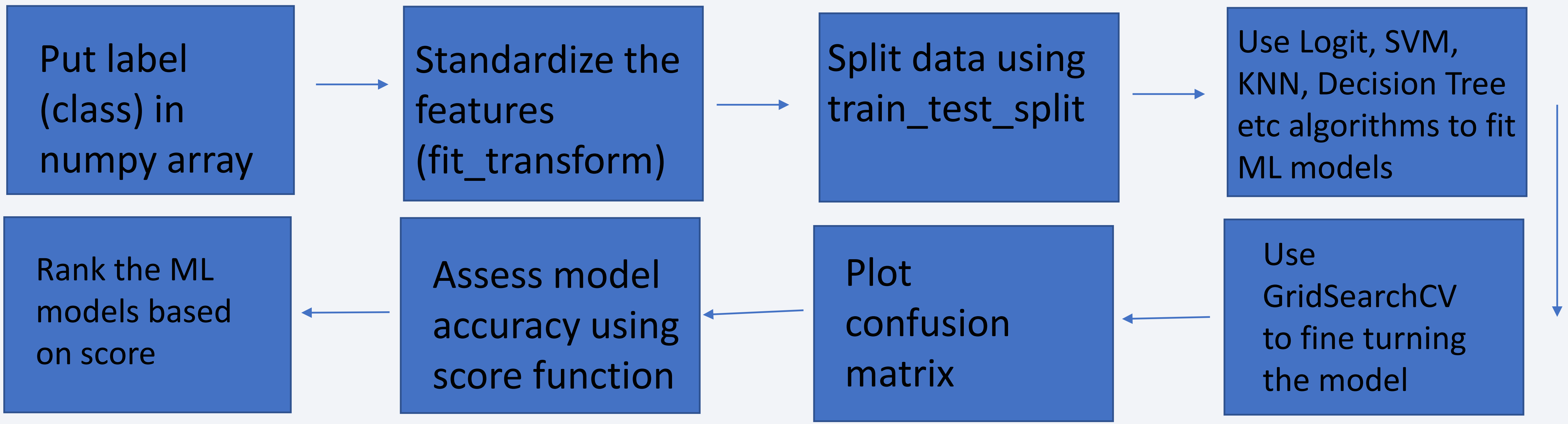
- Summarize what plots/graphs and interactions you have added to a dashboard
  - Add a dropdown list to enable Launch Site selection
  - Add a pie chart to show the total successful launches count for all sites
  - Add a scatter chart to show the correlation between payload and launch success
  - Add a slider to select payload range
- Explain why you added those plots and interactions
  - Assess how important launch site is to success
  - Assess how success rates correlate with pay load

- CapStone-Project/spacex-Launch-Dashboard.py at main · caijiao314159/CapStone-Project (github.com)

GridSearchCV

Predictive Analysis (Classification)

- We imported seaborn & sklearn libraries to be able to build the ML models
- We standardize the data and then split the data into training and testing data.
- We fit models with training set and score the models using testing data.
- We fine tune the model parameters using GridSearchCV.



- CapStone-Project/spacex-ML-Prediction.ipynb at main · caijiao314159/CapStone-Project (github.com)

Results

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

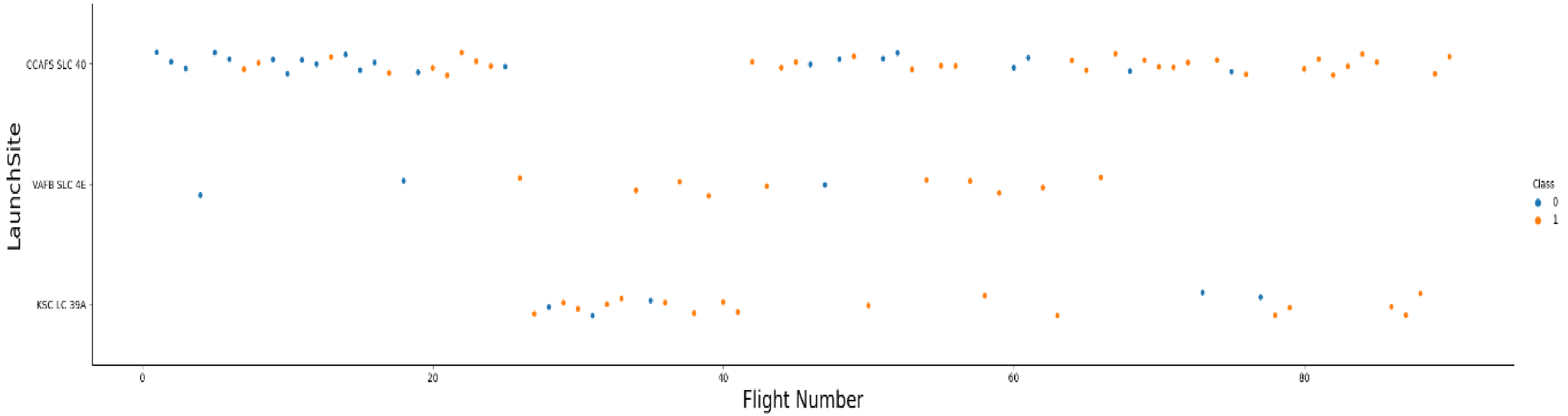
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- The number of flights vary across launch sites.
- Site CCAFS SLC 40 has a lot more launches.

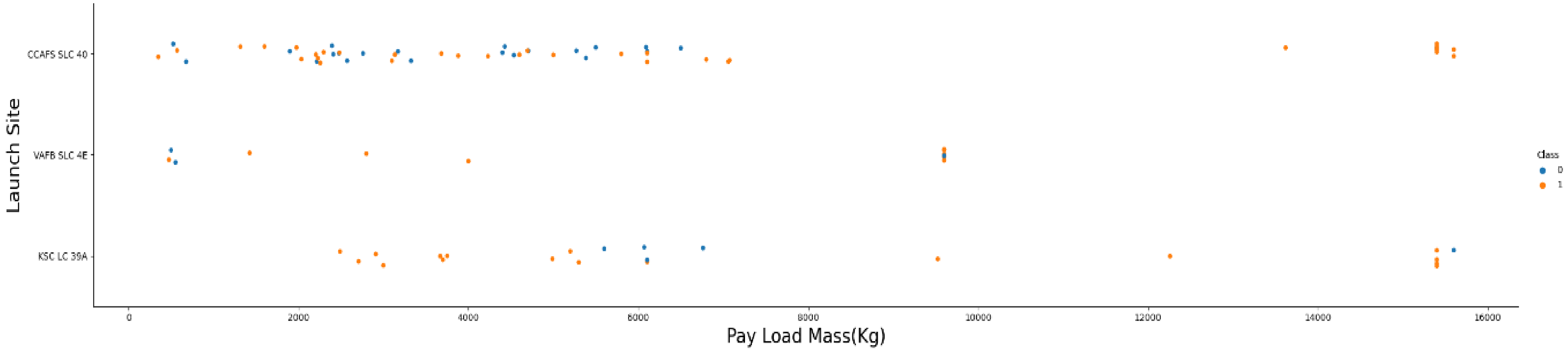
```python
### TASK 1: Visualize the relationship between Flight Number and Launch Site
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Payload vs. Launch Site

- Majority of launches have pay load mass that is between 0 and 7000 kg, but we also see some outliers where the pay load mass is as big as 16,000 kg

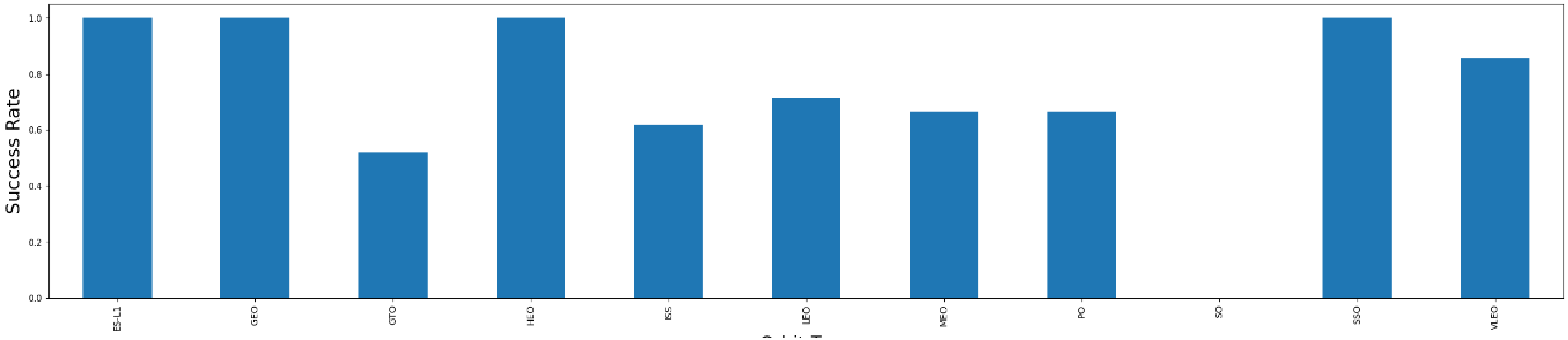
```python
### TASK 2: Visualize the relationship between Payload and Launch Site
sns.catplot(x="PayloadMass",y="LaunchSite",data=df,hue="Class", aspect = 5)
plt.xlabel("Pay Load Mass(Kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Success Rate vs. Orbit Type

- ESL1, GEO, HEO and SSO have the highest success rates (100%)
- SO has % success rate and GTO has 50% success rate.

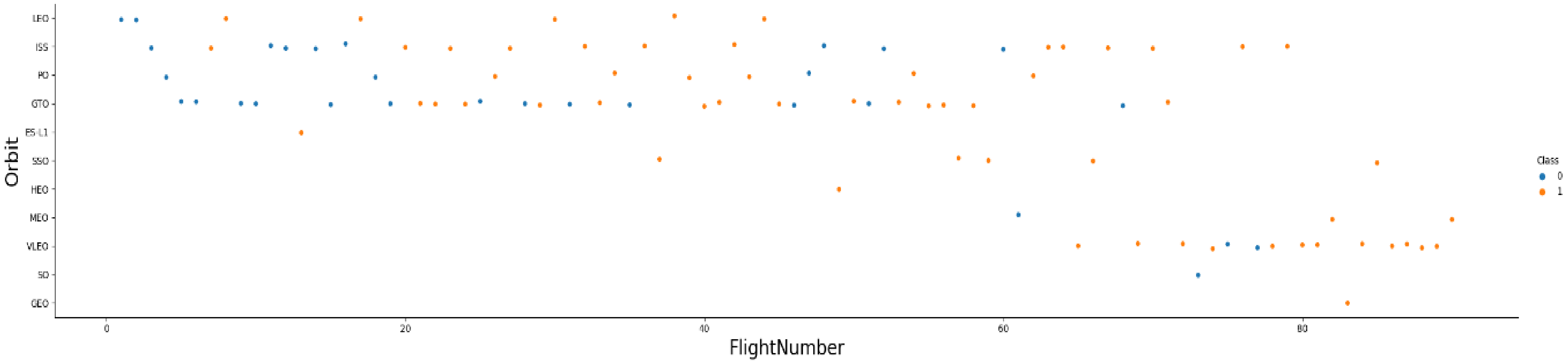
```python
### TASK  3: Visualize the relationship between success rate of each orbit type
df.groupby("Orbit").mean()['Class'].plot(kind='bar')
plt.xlabel("Orbit Type",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



Flight Number vs. Orbit Type

- The higher flight numbers tend to coincide with the orbit types of SO & VLEO and have low success rates.

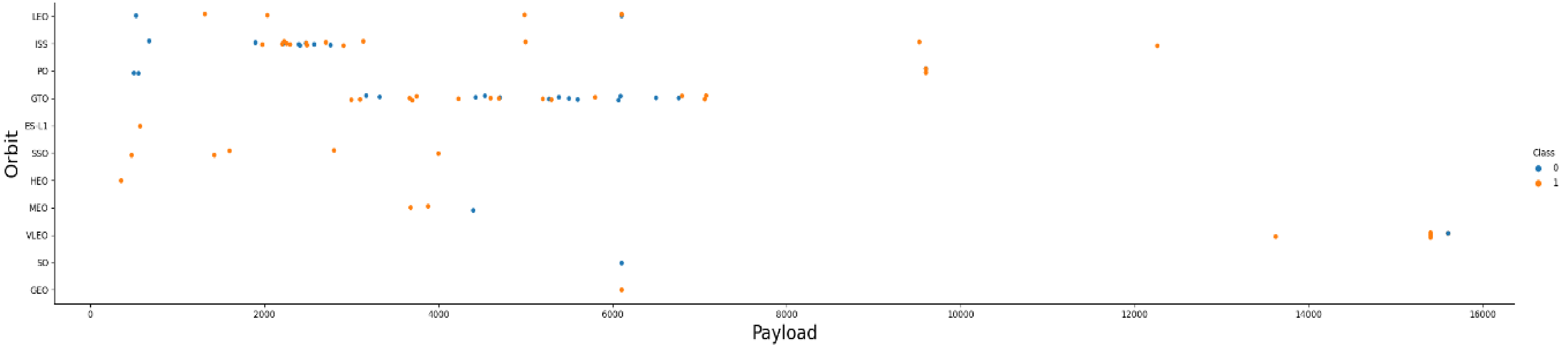
```
### TASK  4: Visualize the relationship between FlightNumber and Orbit type
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Payload vs. Orbit Type

- Launches with low pay load tend to be in ISS orbit; launches with intermediate payloa (between 3500 and 8000 kg) tend to be in GTO orbit

```
### TASK  5: Visualize the relationship between Payload and Orbit type
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

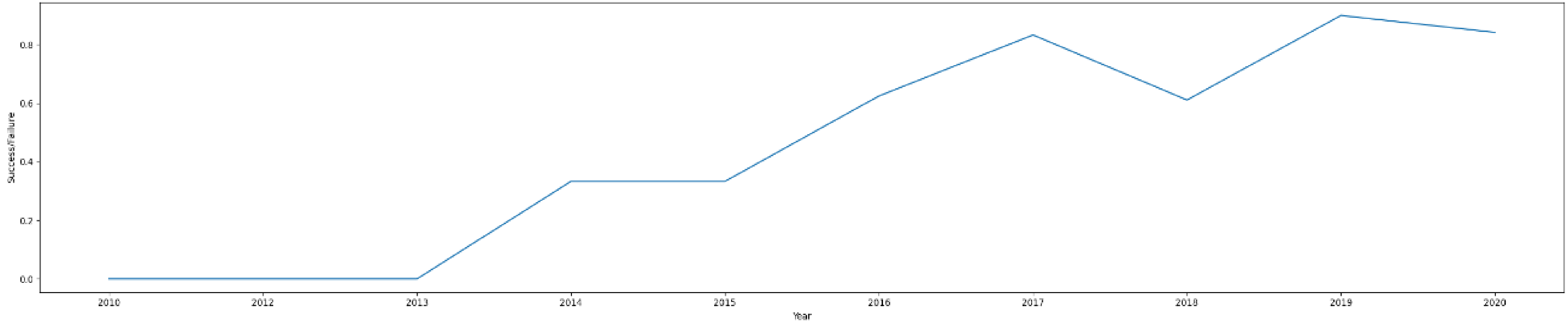


Launch Success Yearly Trend

- Success rate has been increasing since 2013 (with slight dip in 2018).
- The rate of increase was fastest from 2013 to 2017.

```
## Features Engineering
Extract_year()
df['Year'] = year
#df.head()

average_by_year = df.groupby(by="Year").mean()
average_by_year.reset_index(inplace=True)
sns.lineplot(x="Year",y="Class",data = average_by_year)
plt.xlabel("Year")
plt.ylabel("Success/Failure")
plt.show()
```



All Launch Site Names

There are only 4 distinct launch sites: CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, CCAFS SLC-40

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
#df['Launch_Site'].unique()
```

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

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

Launch Site Names Begin with 'CCA'

- The 5 records with launch sites begin with `CCA’ happen to be all for CCAFS LC-40

```
#df['Launch_Site'].loc[df['Launch_Site'].str.startswith('CCA')]
#df['Launch_Site'].unique()

%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- The total payload mass carried by boosters from NASA is 45,596 kg

```
#df['PAYLOAD_MASS__KG_'].loc[df['Customer']=='NASA (CRS)'].sum()
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is 2,928.4 kg

```
#df['PAYLOAD_MASS__KG_'].loc[df['Booster_Version']=='F9 v1.1'].mean()
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version='F9 v1.1';
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- The first successful landing outcome on ground pad was on Dec 22, 2015

```
#df['Date'].loc[df['Landing _Outcome']=='Success (ground pad)'].head(1)
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE [Landing _Outcome]= 'Success (ground pad)';
```

```
19    22-12-2015
Name: Date, dtype: object
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE [Landing _Outcome]= 'Success (drone ship)' AND 4000 < PAYLOAD_MASS__KG_ < 6000;

  - Successful Drone Ship Landing with Payload between 4000 and 6000 include: F9 FT B1021.1, F9 FT B1022, F9 FT B1023.1, F9 FT B1026, F9 FT B1029.1, F9 FT B1021.2, F9 FT B1029.2, F9 FT B1036.1, F9 FT B1038.1, F9 B4 B1041.1, F9 FT B1031.2, F9 B4 B1042.1, F9 B4 B1045.1, F9 B5 B1046.1

```
#df1=df.loc[df['PAYLOAD_MASS__KG_']>4000].loc[df['PAYLOAD_MASS__KG_'
#df1['Booster_Version'].loc[df['Landing _Outcome']=='Success (drone 

%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE [Landing _Outcome]=
```

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

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

- The total number of successful mission outcomes is 100 and the total number of failed mission is

```
#df['Landing _Outcome'].value_counts()
%sql SELECT CASE WHEN MISSION_OUTCOME LIKE 'Success%' then 'Success' else 'Failure' end as Outcome
```

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

Outcome	Total_number
Failure	1
Success	100

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- This needs a subquery and results in 12 distinct Booster versions.

```
%sql SELECT DISTINCT BOOSTER_VERSION
```

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

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- There are 2 launches with “Failure (drone ship)” outcome in 2015.

Landing _Outcome	Booster_Version	Launch_Site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	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
- The outcome ‘No Attempt” is ranked the highest, with 10 launches, followed by “Failure (drone ship)” and “Success (drone ship)”, both at 5.

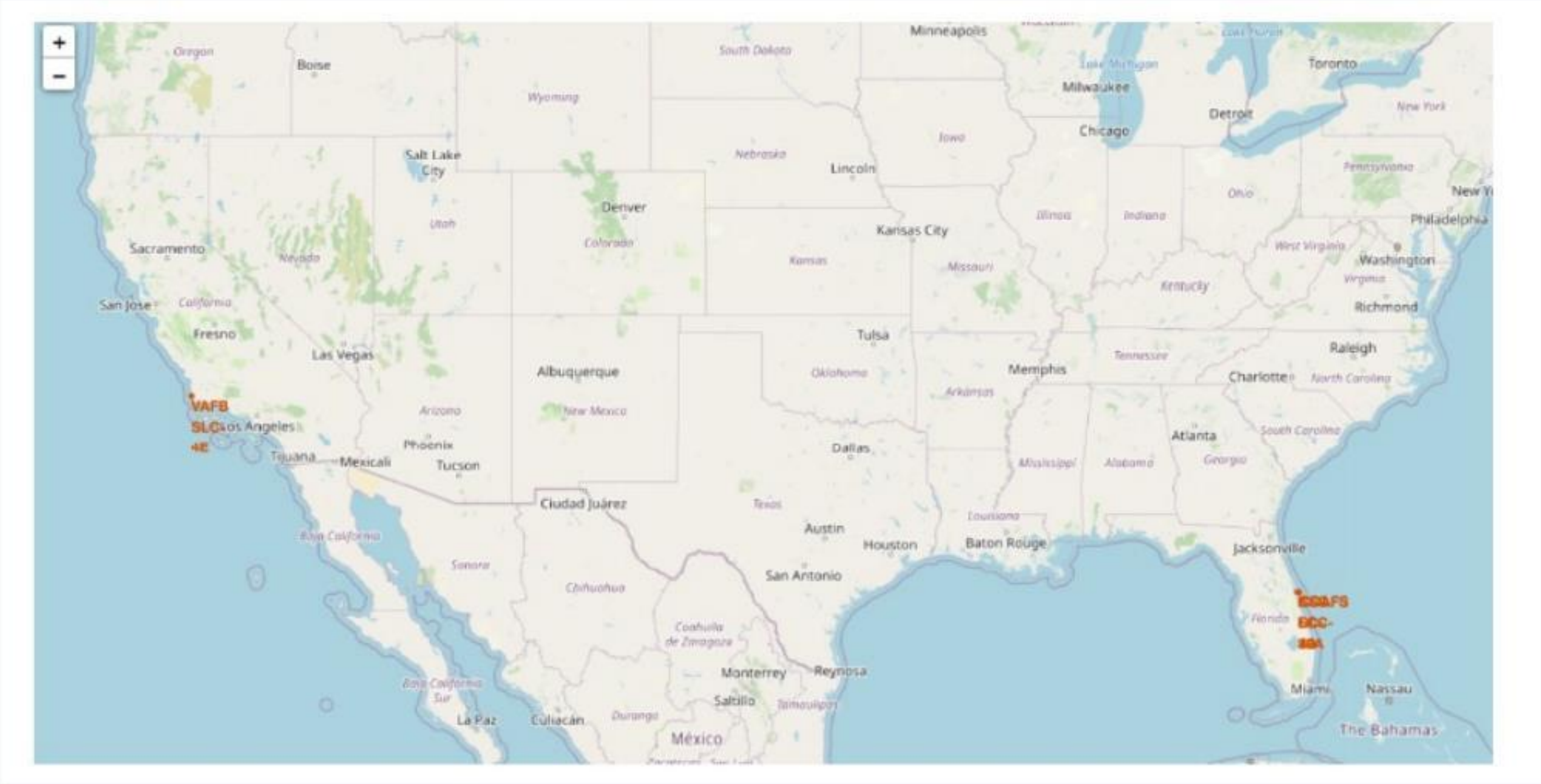
landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

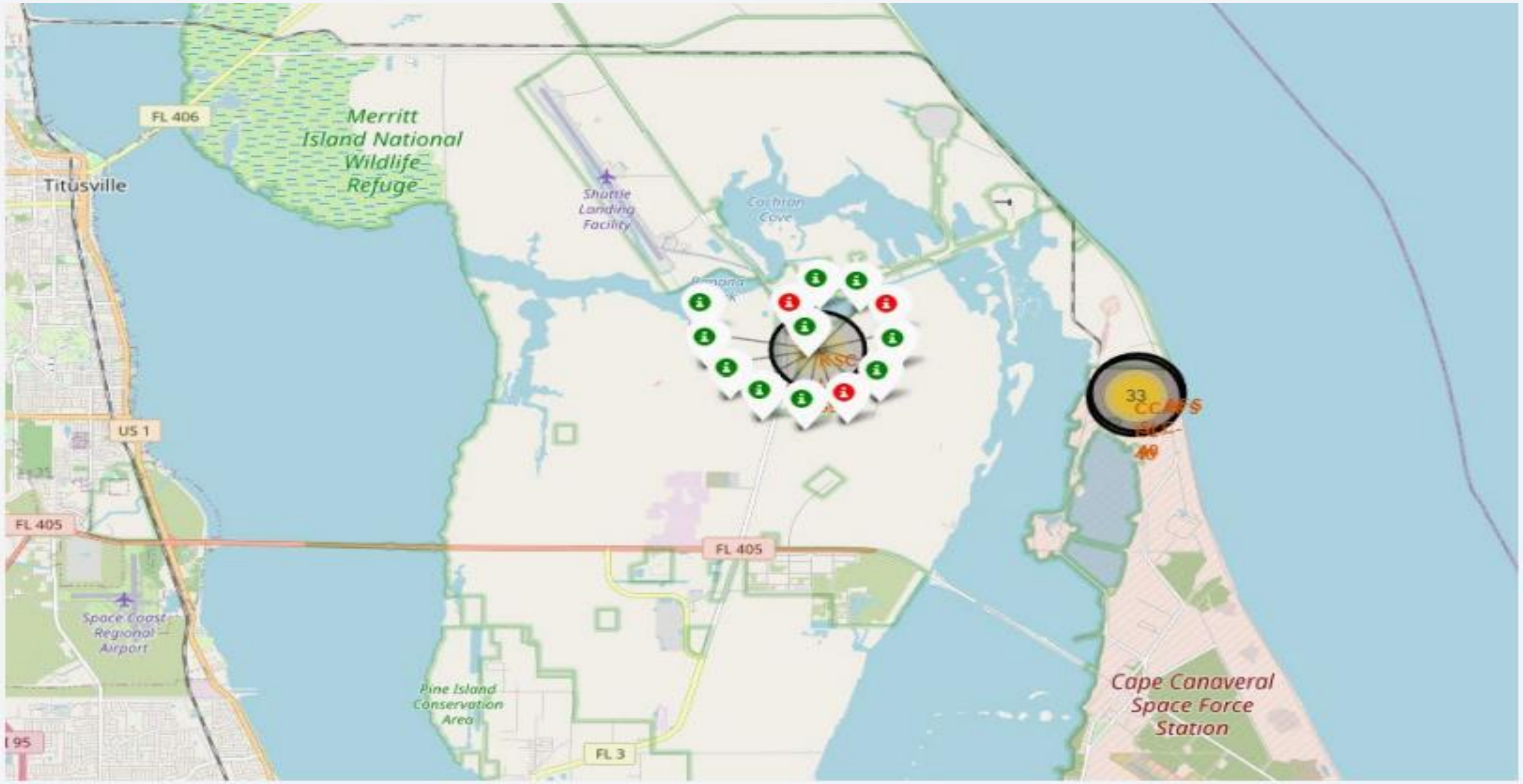
Spacex Launch Sites in US

- Launch Sites tend to be along the coast and southern end of the nation.



Successful and Failed Landings at KSC LC-39A

- The Folium map can be enlarged to display successful and failed landing, color coded (green for success and red for failure).
- Launch site KSC LC-39A has 3 failed and 10 successful landings



Distance of Launch Site to Ocean

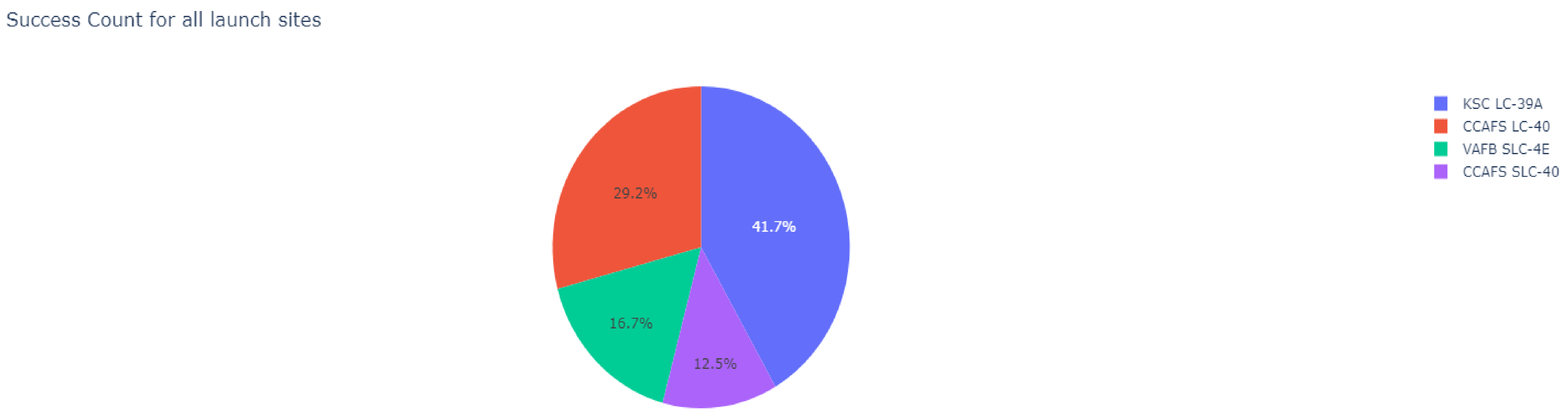
- The launch sites tend to be along the coastlines.
- Failed launches over the ocean result in less damage.

Section 4

Build a Dashboard with Plotly Dash

Distribution of Success Counts across Launch Sites

- KSC LC-39A has the highest success count among all launch sites

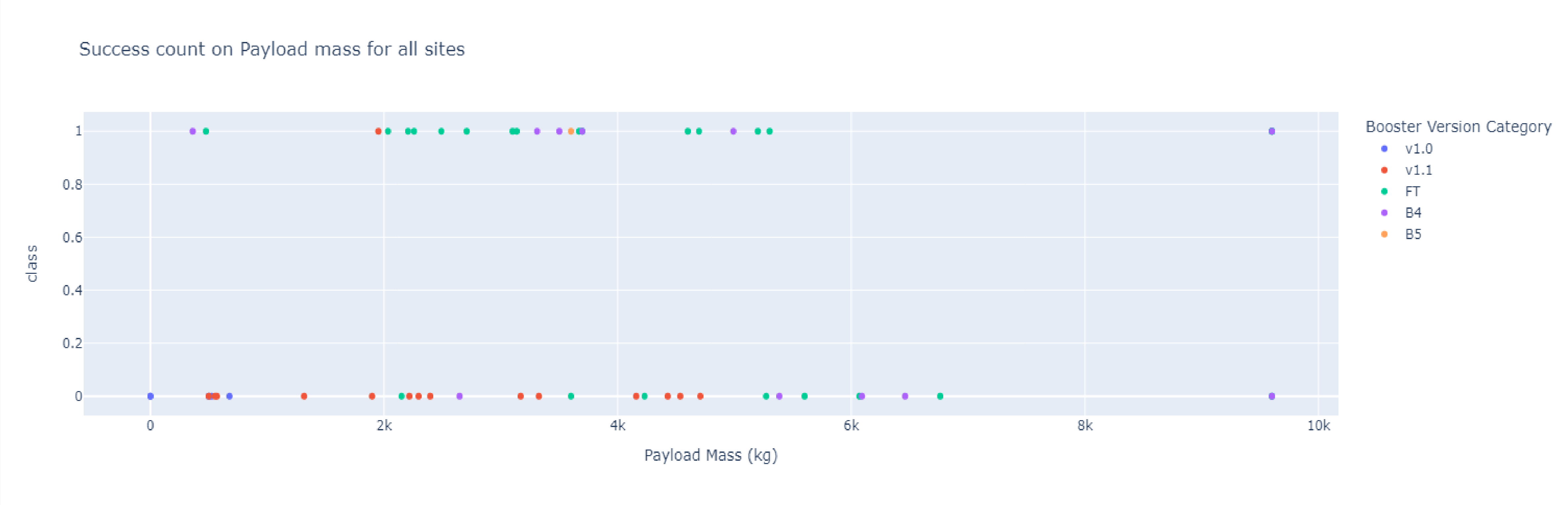


Success Rates for CCAFS SLC-40

- CCAFS SLC-40 has the highest success rate (at 42.9%) among all launch sites

Success rate by Payload and Booster Versions

- On average, Booster Version ‘RT’ has the highest success rate, except when payload Mass gets too high (above 5.5K)
- Booster Version V1.1 has lowest success rate on average

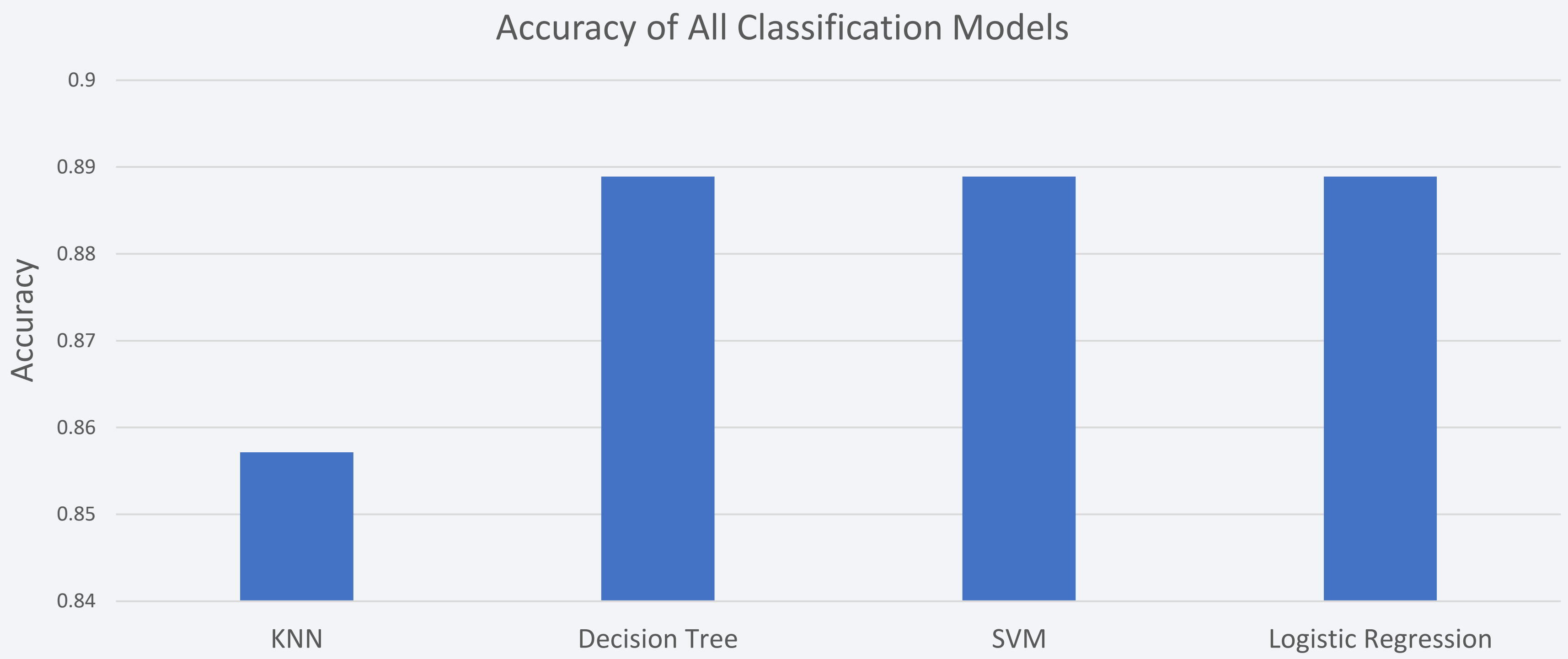


Section 5

Predictive Analysis (Classification)

Classification Accuracy

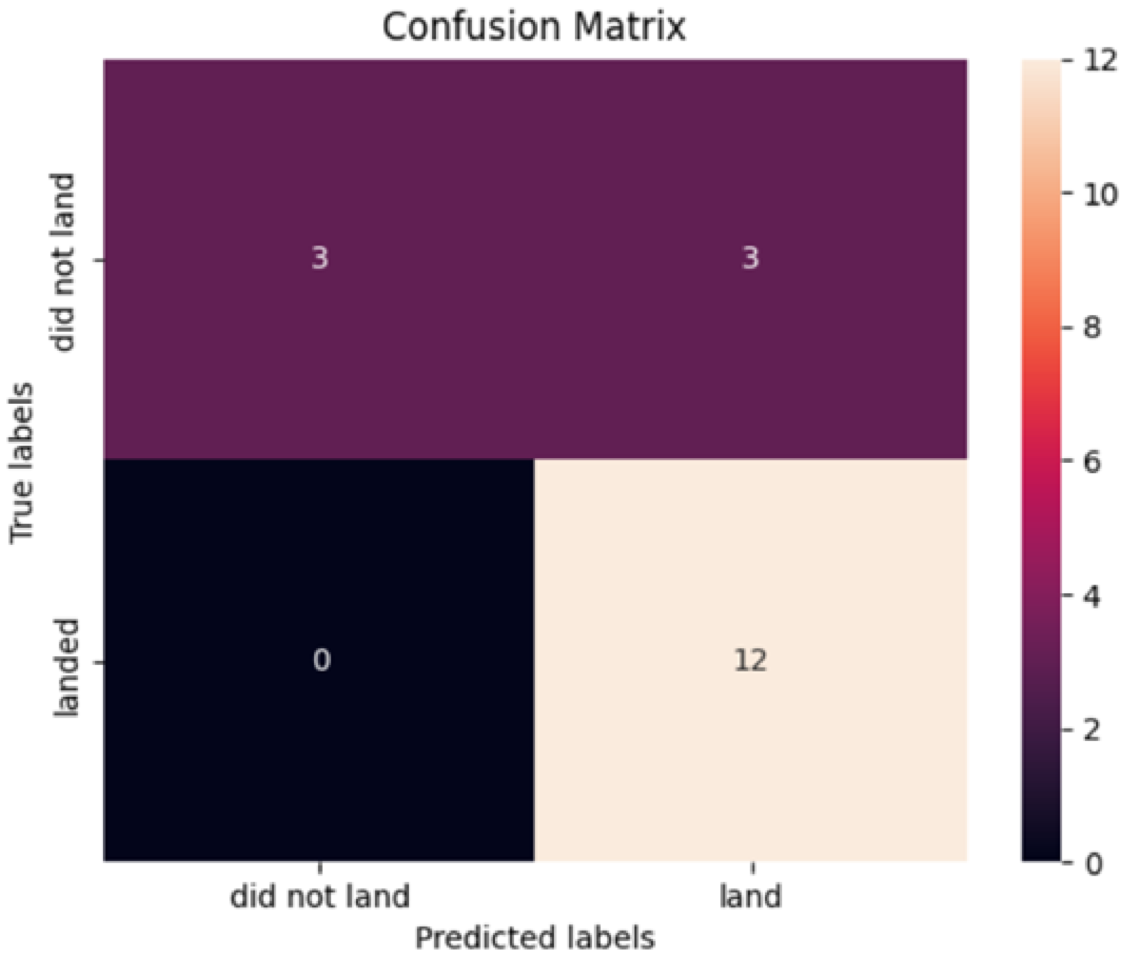
- KNN model has lowest accuracy, while all the other 3 models have very similar scores.



Confusion Matrix

- All 3 models (Logistic Regression, SVM and Decision Tree) produced the same accuracy score and confusion matrix.

- The models predicted the 12 successful landings correctly, but mis-labeled 3 failed landings (false positive).



Conclusions

- Coming up with a ML model to predict the success rate for stage-1 rocket launch is critical to SpaceY's competitiveness.
- We extracted data by using both Spacex public API and web scraping. But the data points we extracted are still quite limited.
- SO & GTO orbit types have low success rates while ESL1, GEO, HEO and SSO orbit types have highest (100%) success rates
- CCAFS SLC-40 launch site has highest success rate.
- On average, Booster Version 'RT' has the highest success rate, while Booster Version V1.1 has lowest success rate.
- With limited data, all 4 classification models (Logistic regression, SVM, KNN and Decision Tree) performed similarly (with KNN being worse).
- False positive is the main problem for model inaccuracy.

Thank you!