IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Xinyu Zhao
06-10-2022

Outline

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

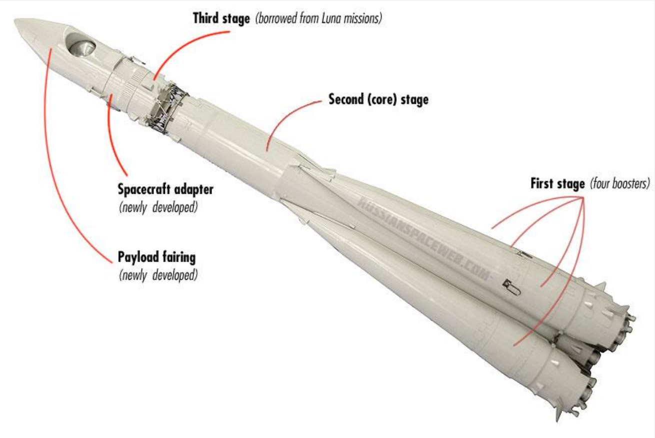
Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Technology advancement has drastically reduced the space launch cost, which give rise to many innovative, privately-owned launch service provider. SpaceX is one such company to revolutionize the space exploration industry. SpaceX's Falcon9 rocket can recover its first stage rocket, the most expensive rocket part, after sending the payload into the orbit. This allows 50% cost-saving compared to non-recoverable rockets.

- However, first stage landing failure sometimes occur and it is therefore critical to explore past data, explore important factors. If we can accurately predict whether the first stage will land, we can determine the cost of a launch. This research will utilize past Falcon9 launch data and apply various visualization, machine learning techniques to give a prediction of Falcon9 landing success rate.



Section 1

Methodology

Methodology

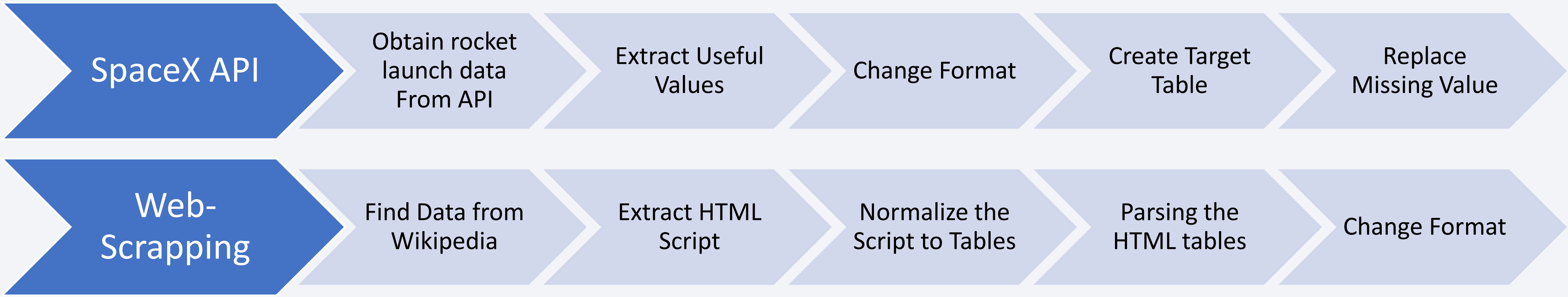
Executive Summary

- Data collection methodology
  - Data Collection Through API / Web-scrapping
- Perform data wrangling
  - Find Patterns and Determine Machine Learning Labels
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Decision Trees, KNN Model, SVM, Logistic Regression

Data Collection

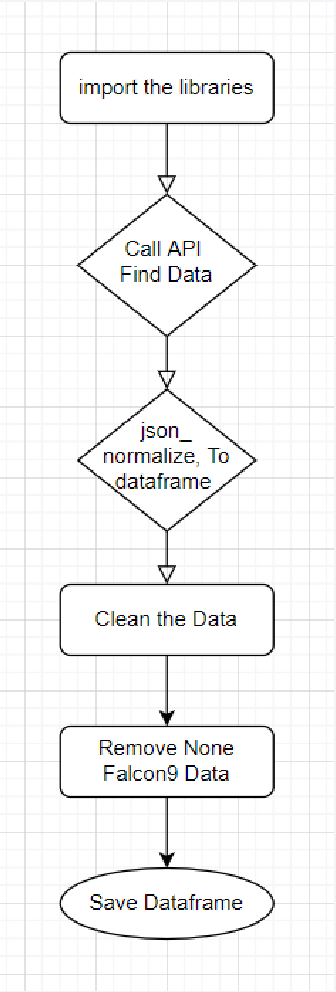
Data is collected through two methods: Web-scrapping & SpaceX API

- SpaceX REST API is an open source API for SpaceX launch data. Using GET method to pull data from the API saves data-processing time because the data is well-structured.
- Web-scrapping allows us to collect data from any website, but the data is mainly unstructured and the cleaning process can be laborious. Wikipedia data table will be web-scrapped and transformed to clean data file.



Data Collection – SpaceX API

- We will first build multiple functions to call required useful data from SpaceX API
- The data downloaded from the API is JSON file, which should be normalized
- Some column in the data are irrelevant IDs, we need to drop them
- We are interested in Falcon9 data but the data also include other boost version like Falcon1, which should be dropped
- Clean the Data and Save to a CSV File



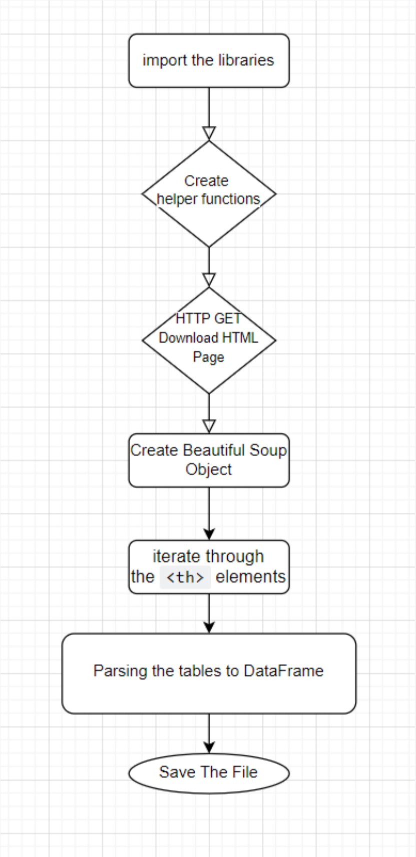
- https://github.com/RamenNoodleJerry/Applied_Data_Science_Capstone/blob/efe6186b58092ba69852caa9ef76dd1601436a0c/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- We will use HTTP GET method to get all HTML script from Wikipedia Falcon9 web page.
- Create a Beautiful Soup object that contain all HTML element of that web page.
- Use “Soup.find_all('table')” to locate our target table
- Iterate through the table’s <th> element and convert the result to Panda DataFrame.
- Save the DataFrame to a CSV file

```
<table class="wikitable plainrowheaders collapsible" style="width: 100%;">
<tbody><tr>
<th scope="col">Flight No.
</th>
<th scope="col">Date and<br/>time (<a href="/wiki/Coordinated_Universal_Time" title="Coordinated Universal Time">UTC</a>)
</th>
<th scope="col"><a href="/wiki/List_of_Falcon_9_first-stage_boosters" title="List of Falcon 9 first-stage boosters">Version,
br/>Booster</a> <sup class="reference" id="cite_ref-booster_11-0"><a href="#cite_note-booster-11">[b]</a></sup>
</th>
<th scope="col">Launch site
</th>
<th scope="col">Payload<sup class="reference" id="cite_ref-Dragon_12-0"><a href="#cite_note-Dragon-12">[c]</a></sup>
</th>
<th scope="col">Payload mass
</th>
<th scope="col">Orbit
</th>
<th scope="col">Customer
</th>
```

	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	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



- https://github.com/RamenNoodleJerry/Applied_Data_Science_Capstone/blob/efe6186b58092ba69852caa9ef76dd1601436a0c/jupyter-labs-webscraping.ipynb

Data Wrangling

- We will use Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- Top 3 SPACEX launch sites:

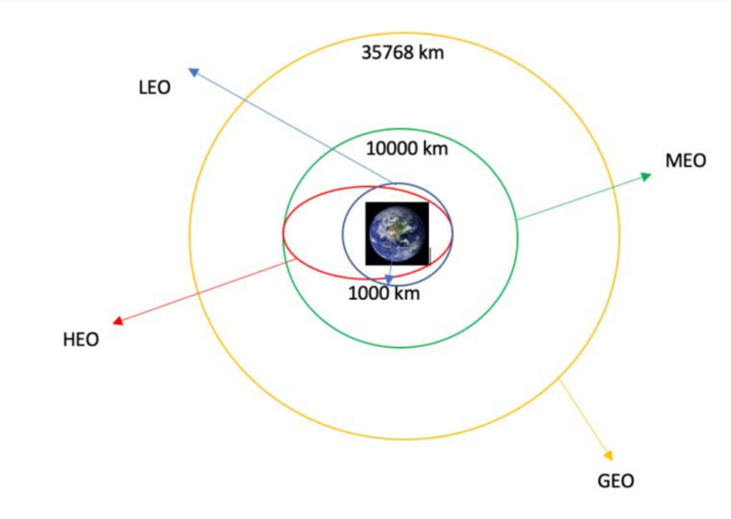
  CCAFS SLC 40

  KSC LC 39A

  VAFB SLC 4E
- Top 3 Launch Orbit Types:

  GTO, ISS, VLEO
- True Ocean, True ASDS, True RTLS represent success. False Ocean, False RTLS, False ASDS, None ASDS, None None represent failure.
- Identify Success and Failure Types and create a landing outcome label called “Class” with 1 for success and 0 for failure. This label will help supervised machine learning prediction

- https://github.com/RamenNoodleJerry/Applied_Data_Science_Capstone/blob/e015834ffb8b6d0708f2228392b8617a8cebd798/labs-jupyter-spacex-Data%20wrangling.ipynb

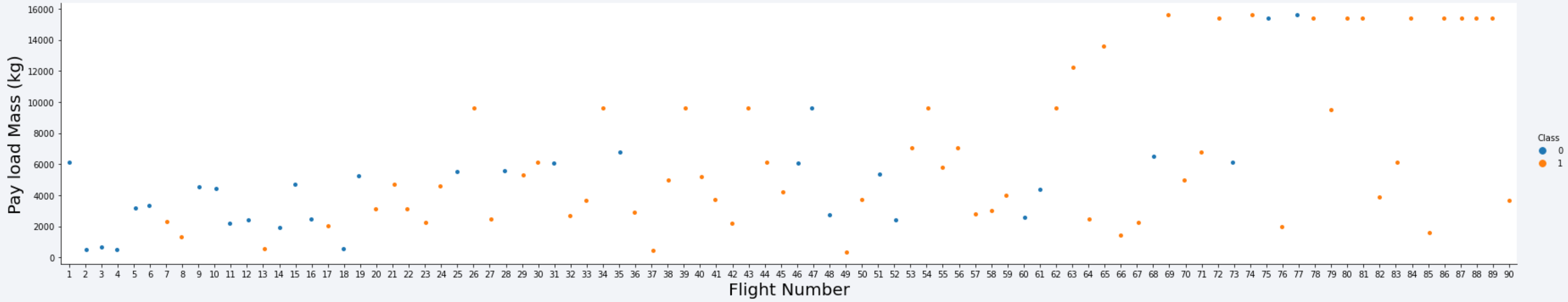



```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

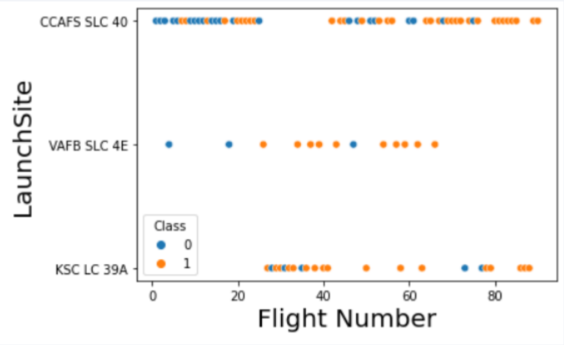
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

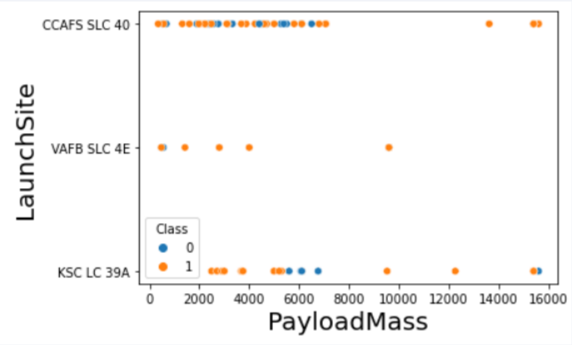
EDA with Data Visualization

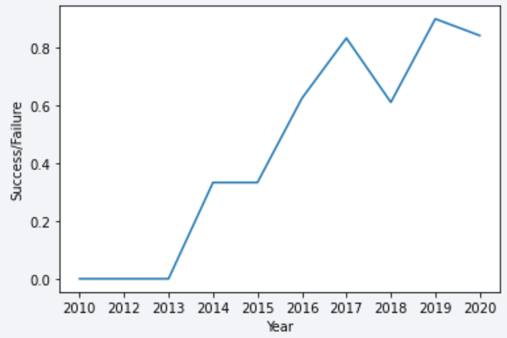
- To better understand our data, we plotted a few graph. First we plotted a catplot that use Flight Number (indicating the continuous launch attempts.) as X-axis, and Pay Load Mass as the Y-axis, and use Class Label as the color of points. This visualize the relationship between three variables. We find that higher launch attempts tend to have higher success rate, high pay load mass tend to have higher success rate.



- We then plot Flight Number and Payload Mass history of three launch sites. We can see that CCAFS SLC40 has the most launch and success rate is positively related to flight number and payload mass. Overtime, the overall success rate is increasing.







- https://github.com/RamenNoodleJerry/Applied_Data_Science_Capstone/blob/ae3eb01433289b951161f983ff48d24c794cb491/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- **We conducted some EDA analysis through SQL query. Our finding is:**
- NASA (CRS) has used SpaceX service to launch 45596kg of payload mass to space
- Falcon9 on average carry 2928.4kg of payload to space each launch
- First successful landing outcome in ground pad was achieved in December 2015
- Falcon9 FT B1022, B1026, B1021.2, B1031.2 are four booster version that succeed in drone ship and have payload mass greater than 4000 but less than 6000
- Overall, Falcon 9 only has 1 mission failed in flight, proving its reliability.
- From 2010 April to 2017 March, SpaceX succeeded in landing F9 launch pod 8 times.

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

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

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

```
%%sql
select min(substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2)) as Date from SPACEXTBL
where "Landing _Outcome" like '%ground pad%'
```

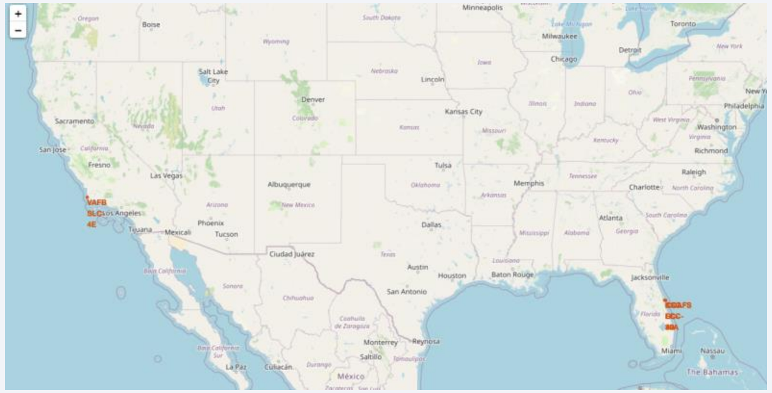
```
 * sqlite:///my_data1.db
Done.
```

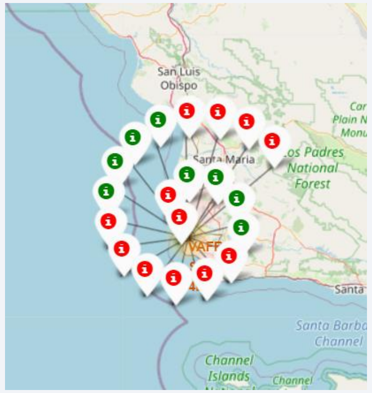
Date
20151222

- https://github.com/RamenNoodleJerry/Applied_Data_Science_Capstone/blob/2c0d17a5e1e5e9b3944d51c7c7082674145ec0f0/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- A folium map allow users to interactively visualize launch sites, success and failure launches, launch sites and proximity on an interactive world map.
- We will first create a Folium map, and mark four launch sites on the map, adding circle and popup label to make them more visible.
- We then add map cluster of past launches and label success/failure of pad recover on each sites (Green is Success, Red is Failure). These data are important that they show the launch result of each sites and understand which one perform well.
- Finally, we mark the distance of launch sites from nearest highway, railway, cities to show how far these sites are from the cities. This can help us determine the launch cost.

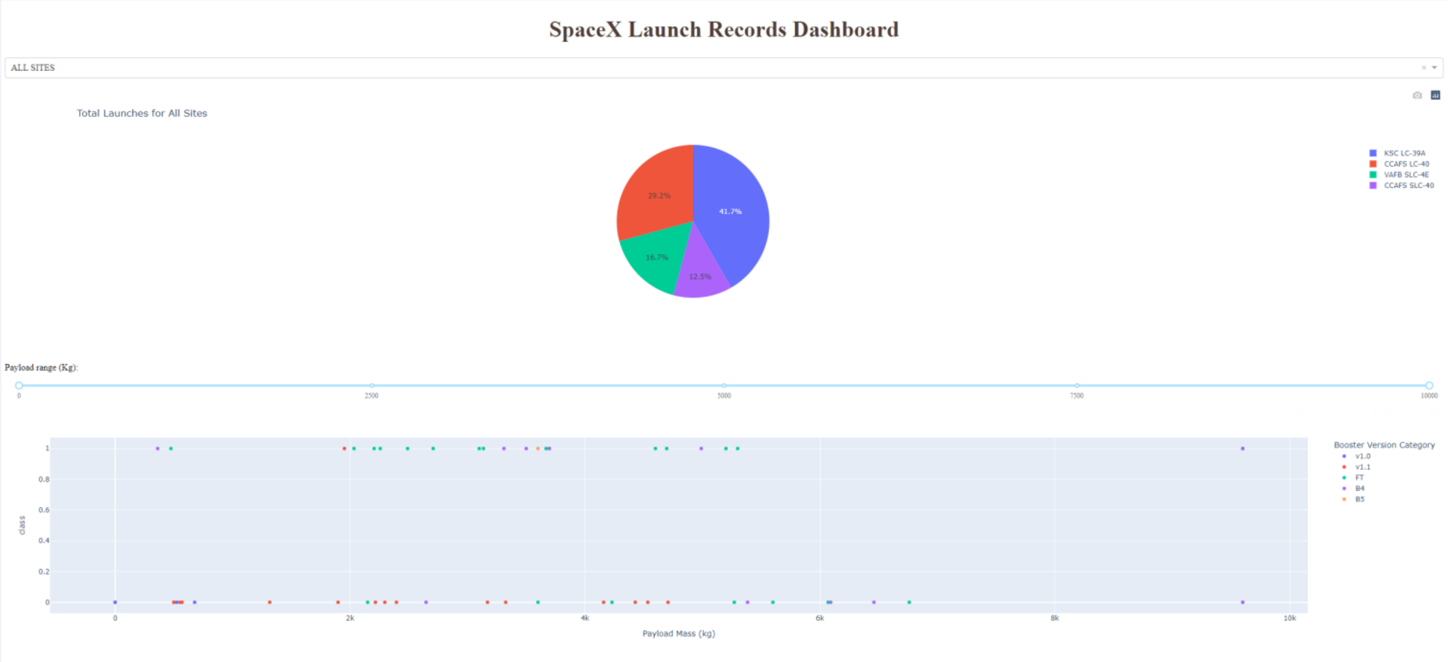




- https://github.com/RamenNoodleJerry/Applied_Data_Science_Capstone/blob/2c0d17a5e1e5e9b3944d51c7c7082674145ec0f0/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

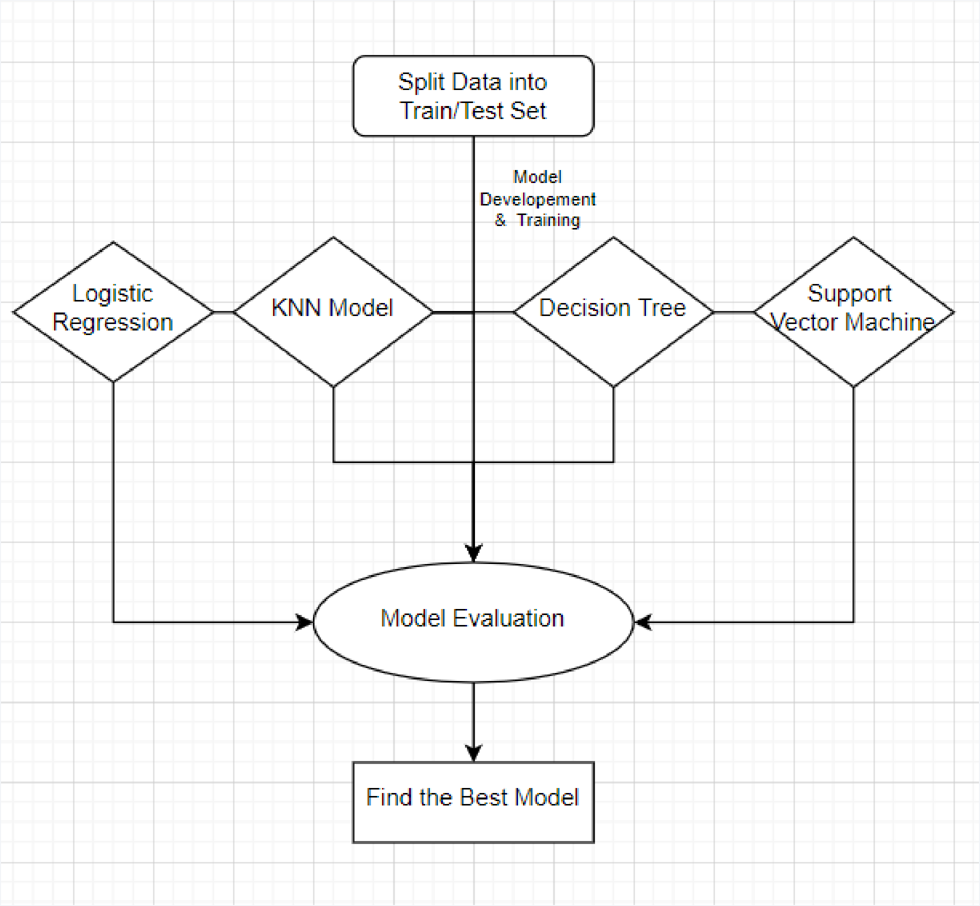
- We create a dashboard using Plotly Dash to interactively show some important charts
- The dashboard starts with a pie chart showing the proportion of launches from different sites. The chart also has a drop down option showing rocket recover Success/Failure rate of each site. This is important as these charts give a wholistic picture of launch sites' overall success rate, which help us to determine the launch cost.
- Following on, a slider of different payload mass and corresponding booster version is shown in a scatter plot. Combined with the pie chart drop down options. The dashboard clearly shows success rate, payload mass, booster version of each launch sites. This gives a wholistic picture of SpaceX launch status.



- https://github.com/RamenNoodleJerry/Applied_Data_Science_Capstone/blob/2c0d17a5e1e5e9b3944d51c7c7082674145ec0f0/spacex_dash_app.py

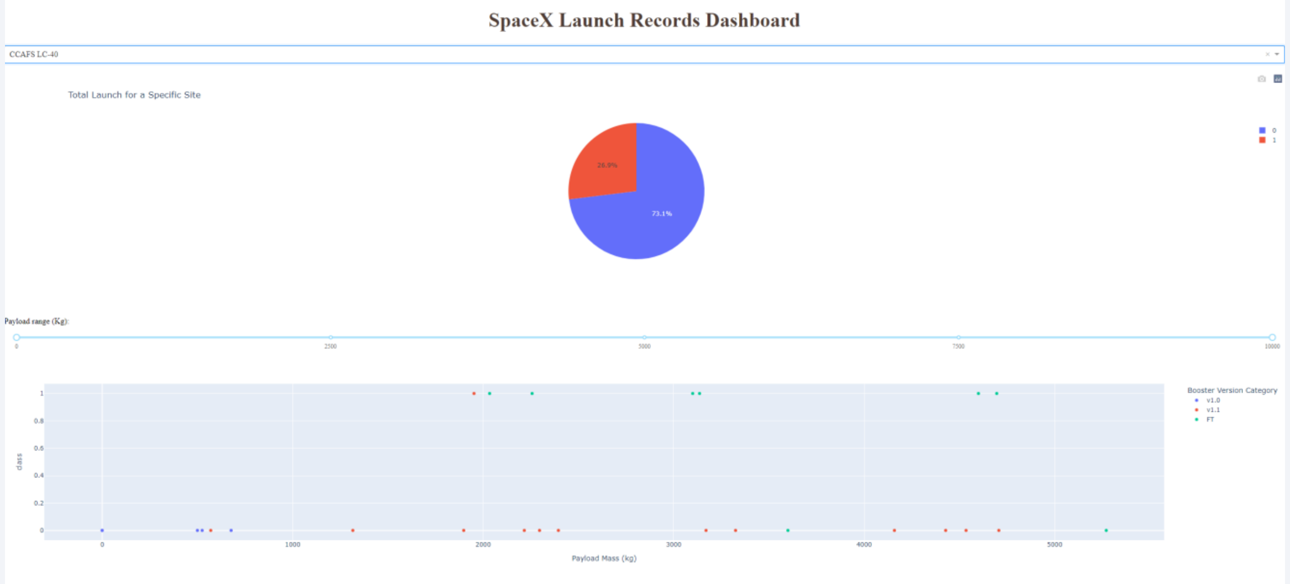
Predictive Analysis (Classification)

- We will conduct classification supervised machine learning method, because our target variable "Class" is labelled and binary.
- We will split data into two groups, one for training, the other for testing. We will develop models using K-Nearest-Neighbor(KNN), Decision Tree, Support Vector Machine, Logistic Regression method. We will then train our model and predict "Class" variable.
- In the end, we will test our model using various model evaluation methods, such as confusion matrix, accuracy score, R^2 score to determine the best model

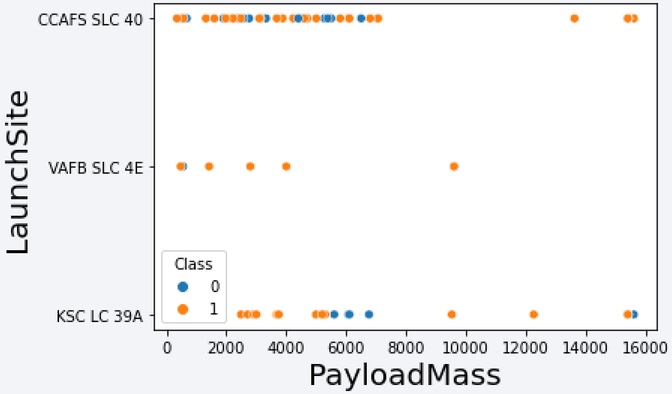


- https://github.com/RamenNoodleJerry/Applied_Data_Science_Capstone/blob/2c0d17a5e1e5e9b3944d51c7c7082674145ec0f0/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results



	Model	R-squared Score
0	KNN	83.333333
1	DecisionTree	83.333333
2	SVM	83.333333
3	LogisticRegression	83.333333



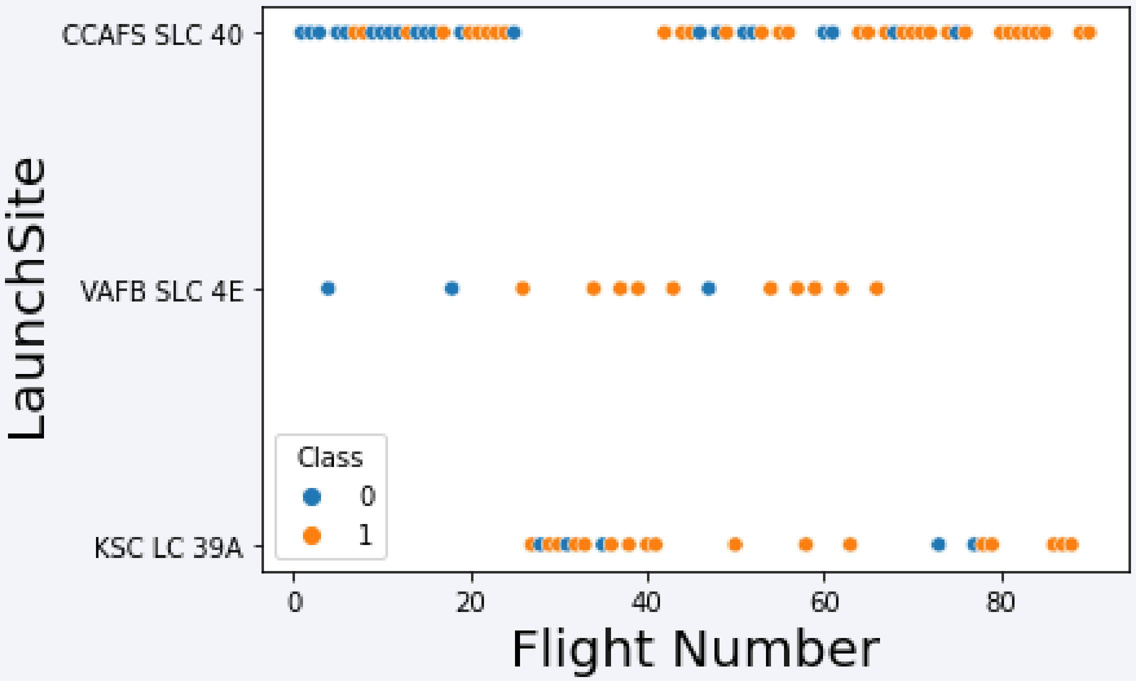
	Model	Accuracy Score
1	DecisionTree	0.875000
0	KNN	0.848214
2	SVM	0.848214
3	LogisticRegression	0.846429

Section 2

Insights drawn from EDA

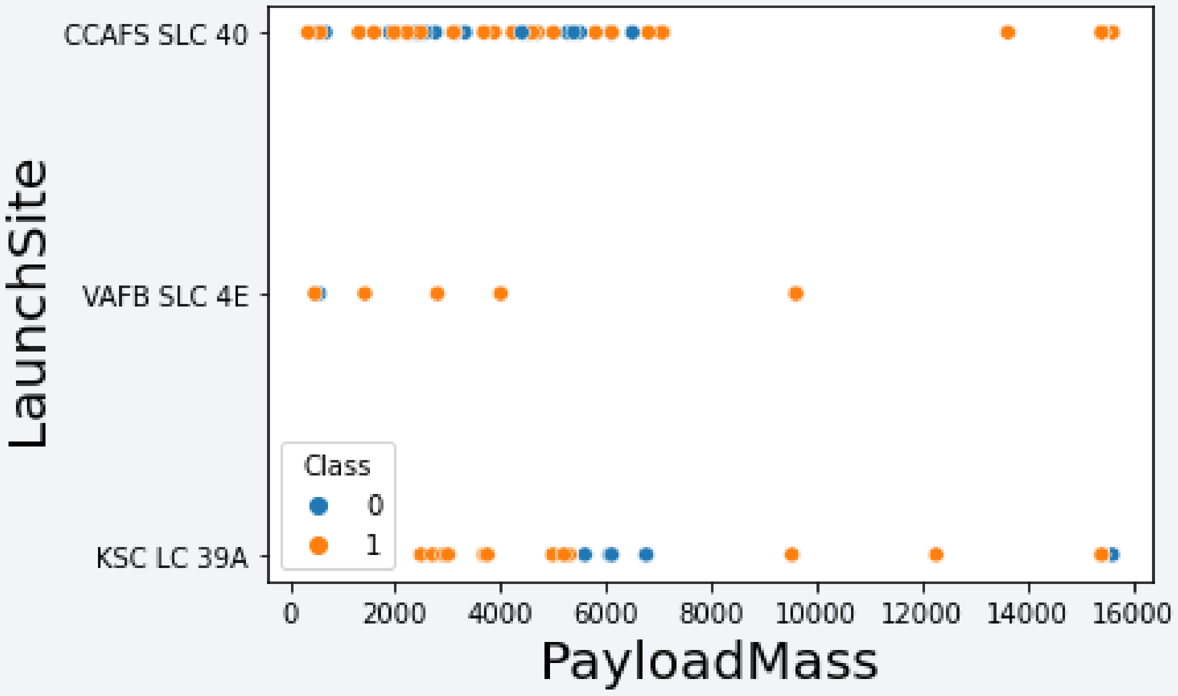
Flight Number vs. Launch Site

- KSC LC 39A success more frequently than the other sites
- CCAFS SLC 40 has a higher frequency of success and the success rate is improving
- The success rate of every sites is improving



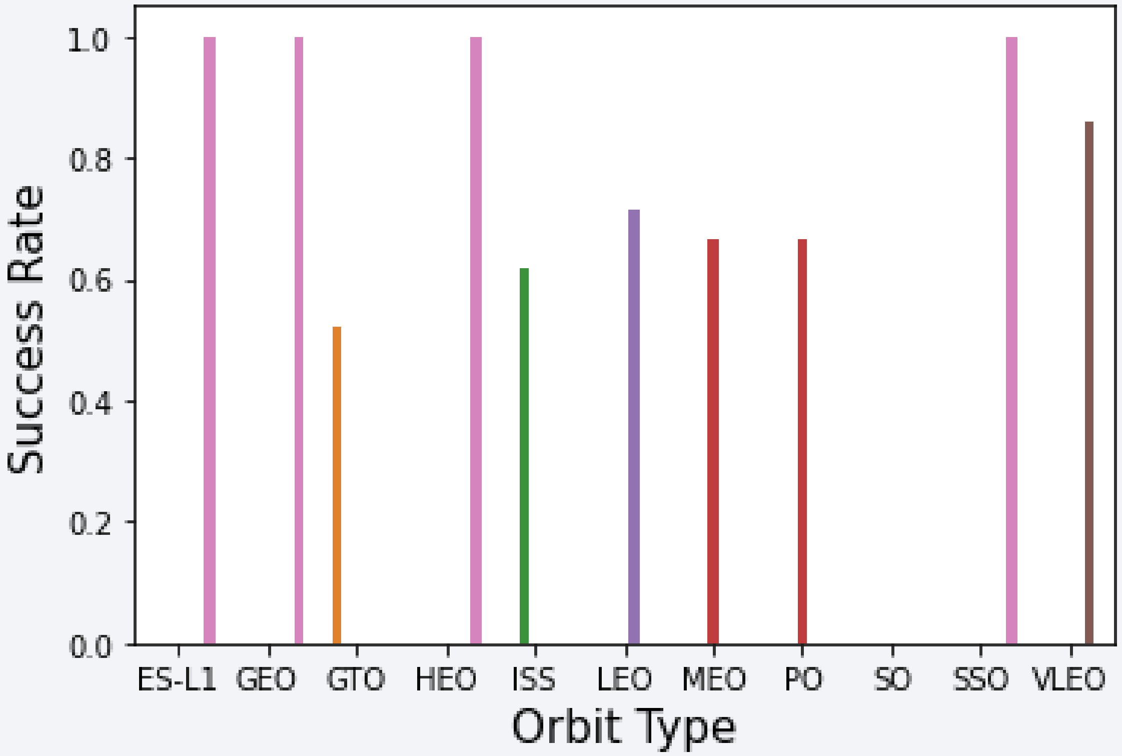
Payload vs. Launch Site

- For the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).
- Success Rate for VAFB SLC 4E is relatively high but there is a fewer launch than the other sites
- Success rate for high payload mass launch is higher



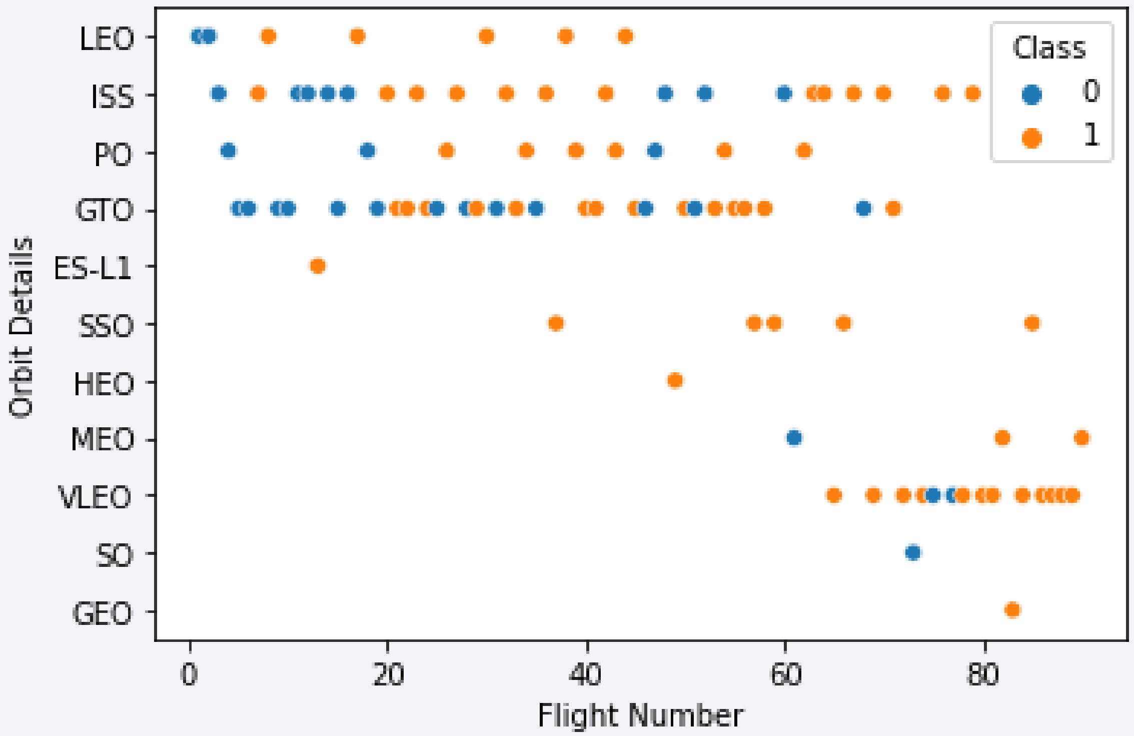
Success Rate vs. Orbit Type

- Launch to ES-L1, GEO, HEO, SSO orbits are the highest while launch to GTO orbit is the lowest
- Launch to SO orbit all failed
- Launch to ISS, LEO, MEO, PO have around 70% success rate.



Flight Number vs. Orbit Type

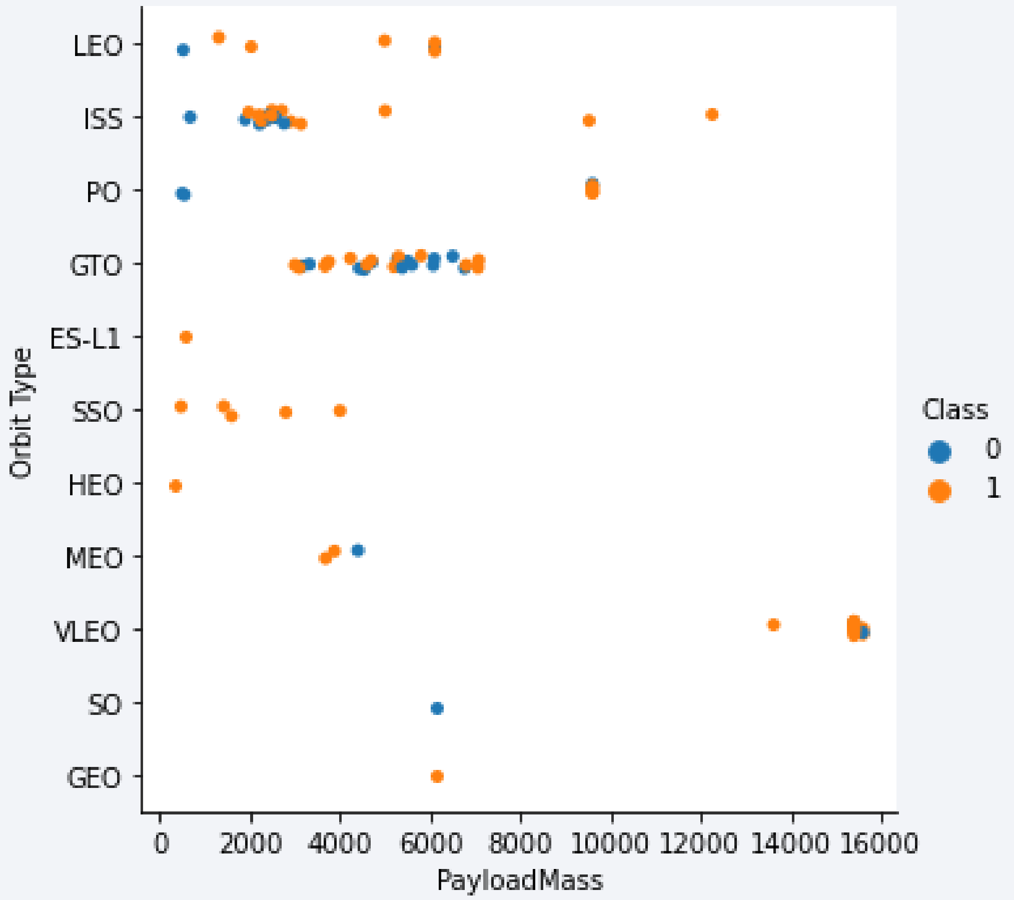
- In the LEO orbit the Success appears related to the number of flights
- There is no relationship between flight number when in GTO orbit
- Flight to ES-L1, SSO, HEO, GEO all succeed
- Flight to SO all failed
- Success to MEO orbit is related to number of flights but the data may be too few



Payload vs. Orbit Type

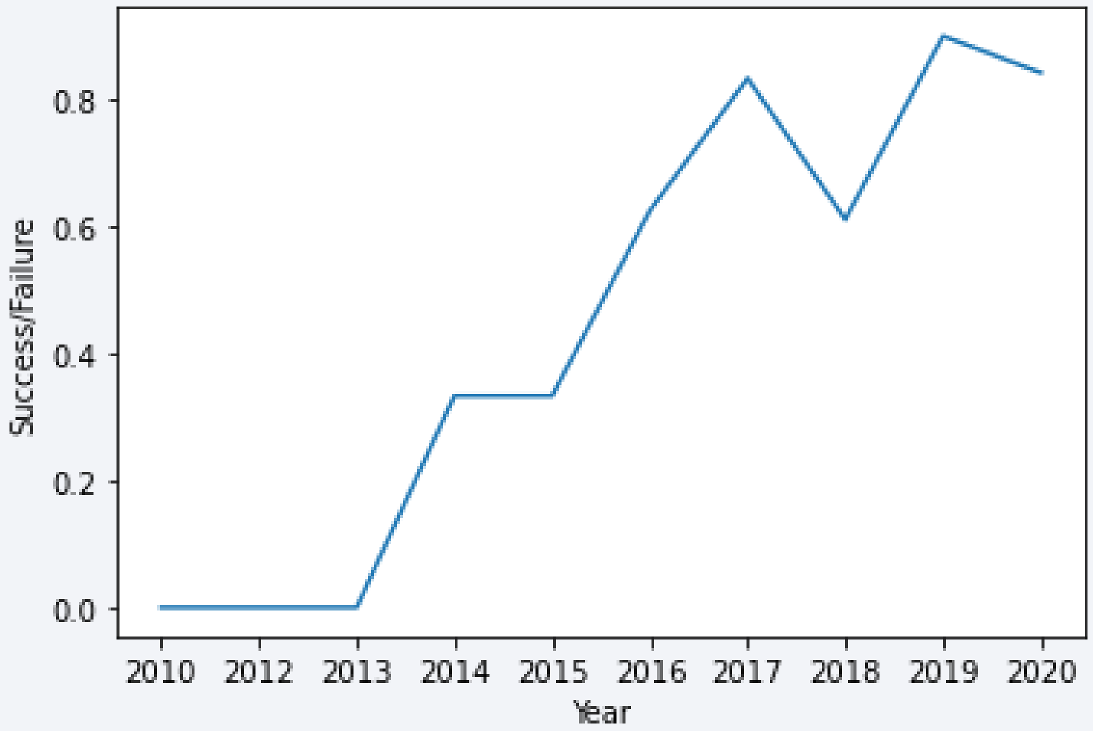
- With heavy payloads the successful landing or positive landing rate are more for PO, LEO and ISS

- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there



Launch Success Yearly Trend

- the success rate since 2013 kept increasing till 2020



All Launch Site Names

- Four unique launch sites are found from record

```
%%sql
select distinct Launch_Site from SPACEXTBL
```

```
 * 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'

- 5 records where launch sites begin with `CCA`

```
%%sql
select * from SPACEXTBL
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
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- NASA has sent 45596 KG of payload mass to space using SpaceX service

```
%%sql
select sum(PAYLOAD_MASS__KG_) as 'Total Payload Mass' from SPACEXTBL
where Customer = 'NASA (CRS)'
```

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

Total Payload Mass
45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1 is 2928.4 KG per launch

```
%%sql
SELECT avg(payload_mass__kg_) as average_payload
from SPACEXTBL where (booster_version) = 'F9 v1.1'
```

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

average_payload
2928.4

First Successful Ground Landing Date

- 2015 December 22nd is the frist successful ground landing.
- We use substring method as SQLite does not have date format (We extract year, month, day separately and combine them to a new string)
- We use “Landing _Outcome” for SQLite to recognize

```
%%sql
select min(substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2)) as Date from SPACEXTBL
where "Landing _Outcome" like '%ground pad%'
```

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

Date
20151222

Successful Drone Ship Landing with Payload between 4000 and 6000

- 4 landings satisfy the requirements

```
%%sql
select BOOSTER_VERSION from SPACEXTBL
where "Landing _Outcome"='Success (drone ship)'
and PAYLOAD_MASS__KG_ BETWEEN 4001 and 5999
```

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

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

Total Number of Successful and Failure Mission Outcomes

- 100 missions are successful, 1 mission failed in flight

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

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

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

Boosters Carried Maximum Payload

- 12 boosters carry maximum payload

```
%%sql
select BOOSTER_VERSION from SPACEXTBL
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
 * 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

- 2 landings failed in 2015

```
%%sql
select substr(Date,4,2) as month, "Landing _Outcome", BOOSTER_VERSION, Launch_Site
from SPACEXTBL
where "Landing _Outcome" like '%Failure%' and substr(Date,7,4) = '2015'
```

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

month	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

- During these periods, 10 landing outcomes are no attempt
- Roughly equal failure and success rate

```
%%sql
select "Landing _Outcome", count(*) as Count_Launches from SPACEXTBL
where substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2) between '20100604' and '20170320'
group by "Landing _Outcome"
order by count(*) desc
```

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

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

Section 3

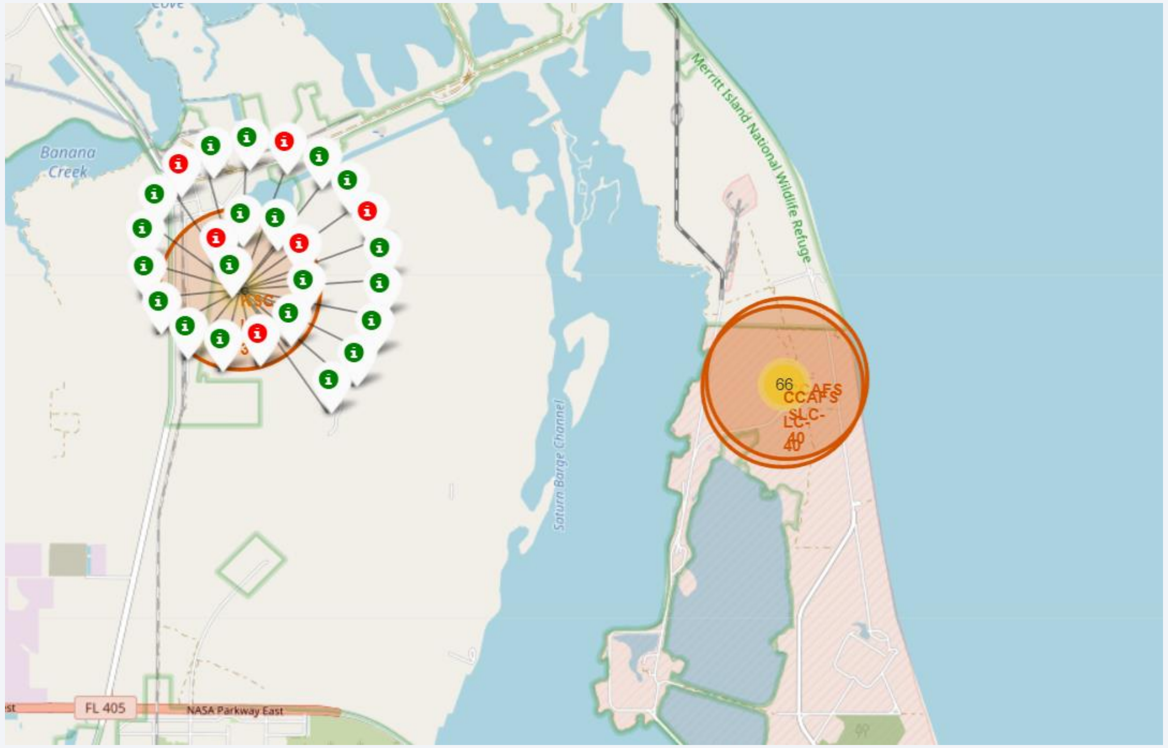
Launch Sites Proximities Analysis

Mark all launch sites on a map

- One launch site locates in California and three launch sites in Florida
- We mark the site with circle and popup label to make them more visible

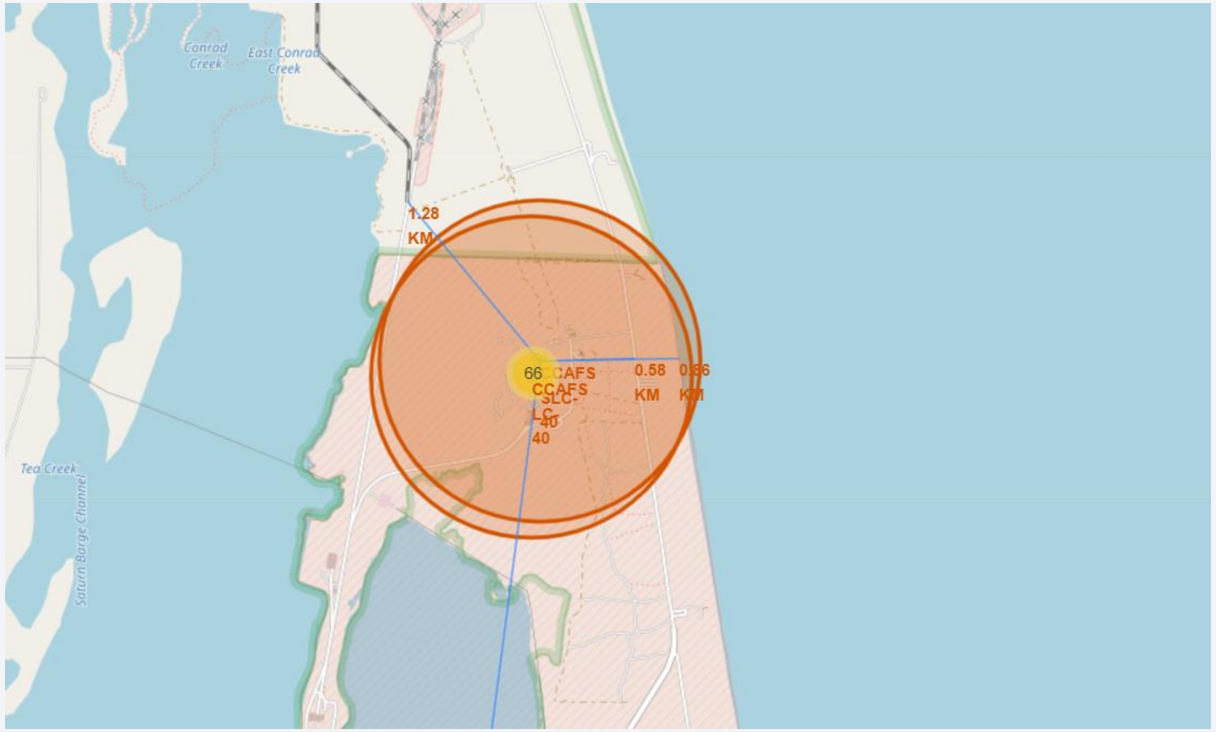
Mark the success/failed for each site on the map

- The mark cluster (yellow number clusters) summarize the success/failed on the map.
- Before click on the cluster, the number shows total launches. When clicked, a circle of green and red mark is shown. Green = Success, Red = Failure



The distances between a launch site to its proximities

- Adding labels and lines to indicate a launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

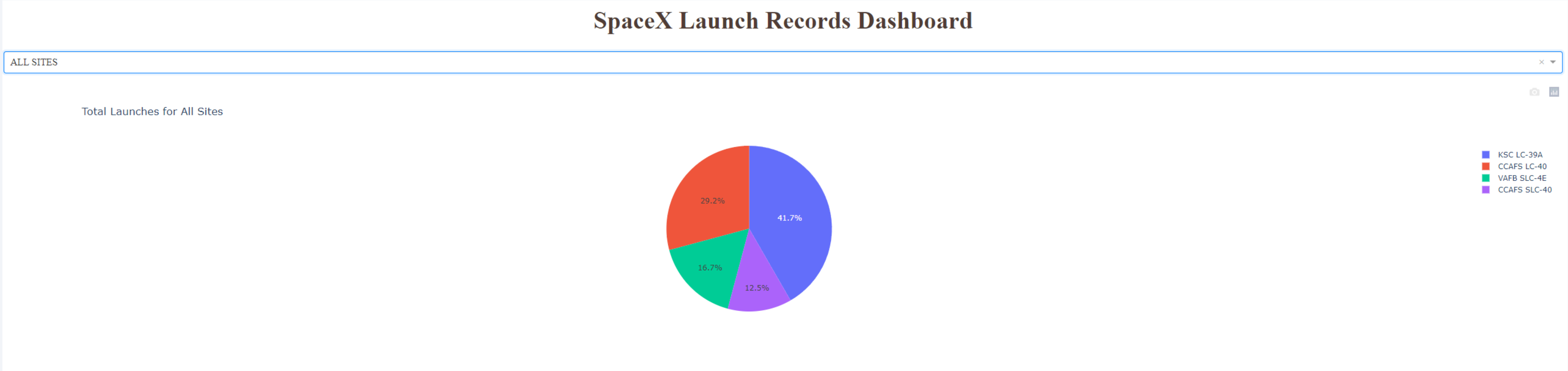


Section 4

Build a Dashboard with Plotly Dash

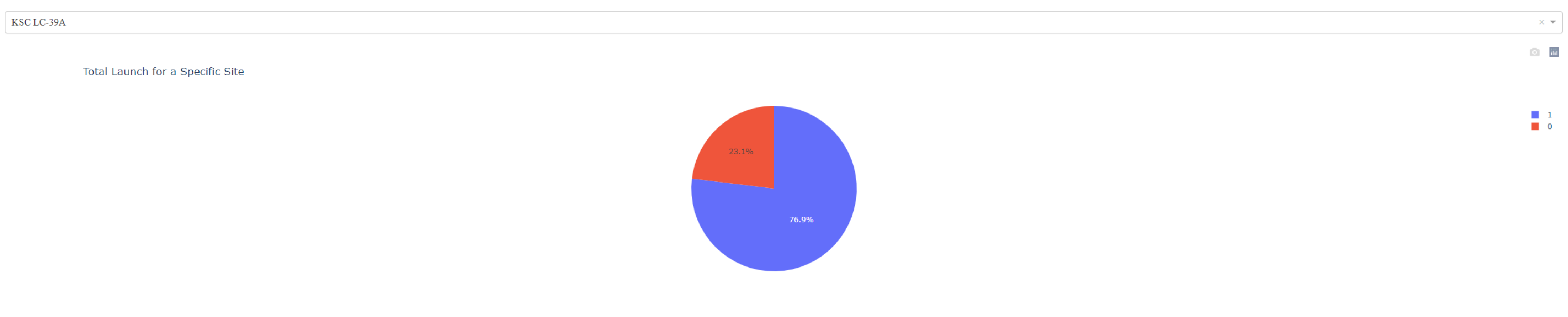
SpaceX Launch Dashboard – Total Launches

- From the dashboard, we can see that 41.7% of launches are from KSC LC-39A
- The pie chart shows the proportion of launches from different launch sites



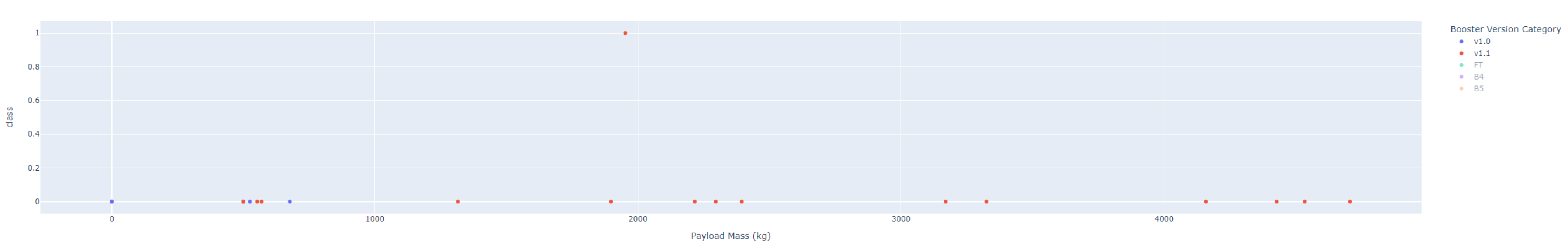
Which Site Has the Highest Success Rate?

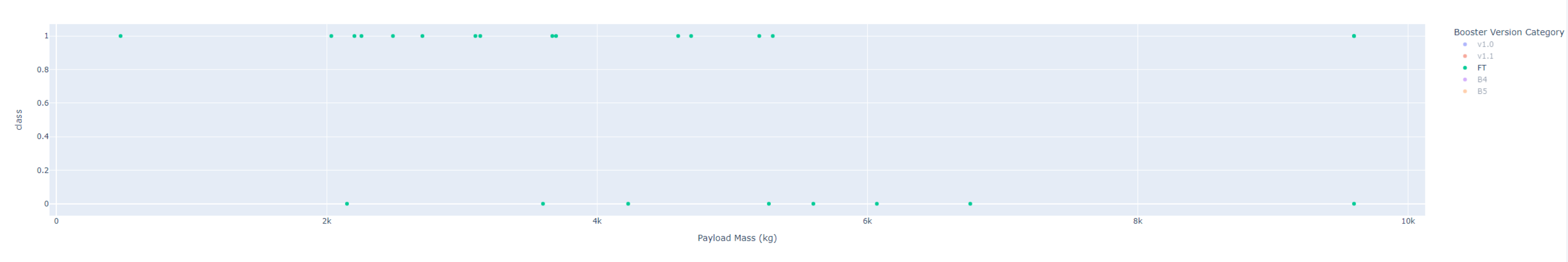
- KSC LC-39A has the highest launch success rate, 76.9%



Success/Fails for Booster Version

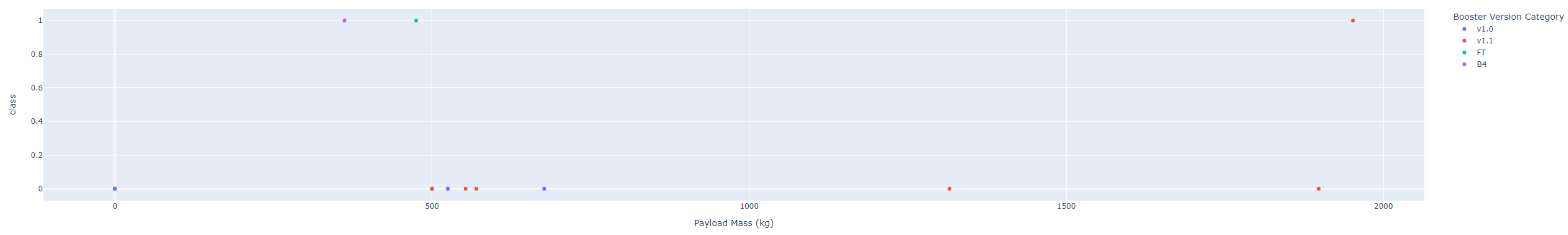
- From the dashboard scatter plot, we can see that success rate for FT is the highest while failed mostly for v1.0 and v1.1

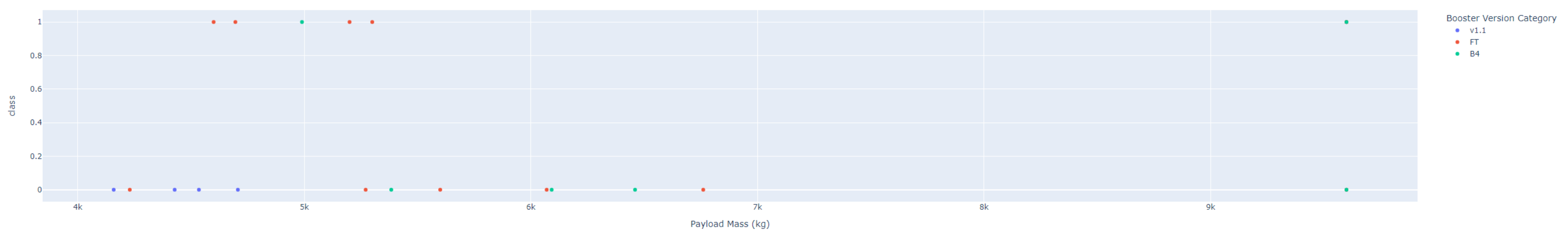


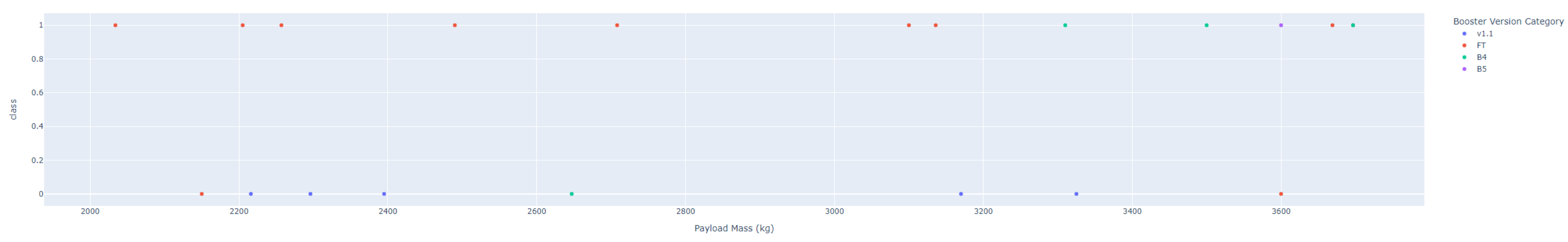


Success/Fails for Payload Mass

- From the dashboard scatter plot, we can see that success rate for payload mass larger than 4000KG and less than 2000KG mostly failed. The success rate is larger than 50% between 2000 and 4000KG





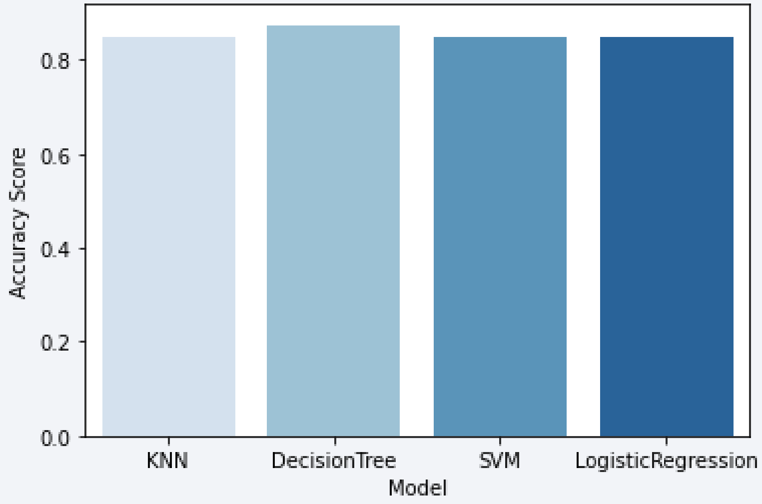


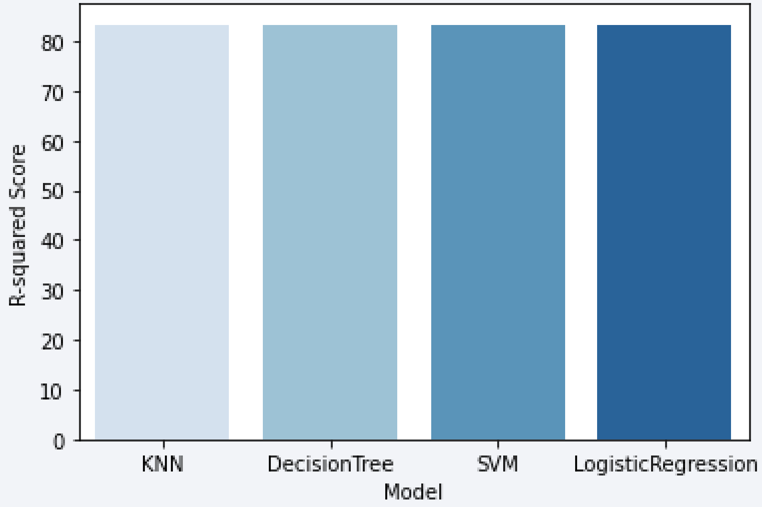
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Out of four models, decision tree has the highest accuracy score.
- All models have the same R-square Score
- Therefore, we choose Decision Tree for our prediction





Confusion Matrix

- The Confusion Matrix of the decision tree model shows that out of the 18 out-of-sample test data, the model predict correctly 15 times.
- The model made 3 mistakes. All of the errors are false success. The model predict successful landings while the true results are failure.

Conclusions

- Launches to ES-L1, GEO, HEO, SSO orbits has the highest success chance
- KSC LC 39A success more frequently and should investigate why
- Success rate for high payload mass launch is higher
- In the LEO orbit the Success appears related to the number of flights
- Overall success rate is improving
- The success rate is larger for payload between 2000KG and 4000KG
- Most missions are successful and should focus on rocket recover success rate
- Although Decision Tree model works the best, there is still chance of false success. A company could suffer intolerable loss if a presuming success launch failed. Therefore we need to consider more complex models to minimize false success error.

Appendix – Dash Callback Code

```
# TASK 4:
# Add a callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output
@app.callback(
    Output(component_id='success-payload-scatter-chart', component_property='figure'),
    [Input(component_id='site-dropdown', component_property='value'),
    Input(component_id='payload-slider', component_property='value')])

def update_graph(site_dropdown, payload_slider):
    if site_dropdown == 'ALL':
        filtered_data = spacex_df[(spacex_df['Payload Mass (kg)']>=payload_slider[0])
        &(spacex_df['Payload Mass (kg)']<=payload_slider[1])]
        scatterplot = px.scatter(data_frame=filtered_data, x="Payload Mass (kg)", y="class",
        color="Booster Version Category")
        return scatterplot
    else:
        specific_df=spacex_df.loc[spacex_df['Launch Site'] == site_dropdown]
        filtered_data = specific_df[(specific_df['Payload Mass (kg)']>=payload_slider[0])
        &(spacex_df['Payload Mass (kg)']<=payload_slider[1])]
        scatterplot = px.scatter(data_frame=filtered_data, x="Payload Mass (kg)", y="class",
        color="Booster Version Category")
        return scatterplot

# Run the app
if __name__ == '__main__':
    app.run_server()
```

Thank you!