IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

J Nash
11/06/23

Outline

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

Executive Summary

- Summary of methodologies
- Data Collection
- Data wrangling
- EDA
- Interactive visual analysis
- Predictive analysis using classification
- Summary of all results
- The features relevant to successful launch and landing would be orbits, reused count, block, payload mass, serial, GridFins and launch sites.
- The optimal launch site is KSC LC-39A
- The best machine learning method to predict successful landings would be the logistic regression model.

Introduction

- Project background and context:
- One of the reason that Space X is successful is that it can decrease the cost dramatically by reusing the first stage of the launch. Space Y wants to competes Space X in the future and thus would like to predict if Space X's Falcon 9 will land successfully based on public information rather than rocket science used by Space X.
- Problems:
- What features are relevant to a successful launch and reuse of the first stage?
- What is the best location for the launch in terms of success rates?
- What is the best machine learning method to predict if a Space X Falcon 9 will land successfully and resue the first stage?

Section 1

Methodology

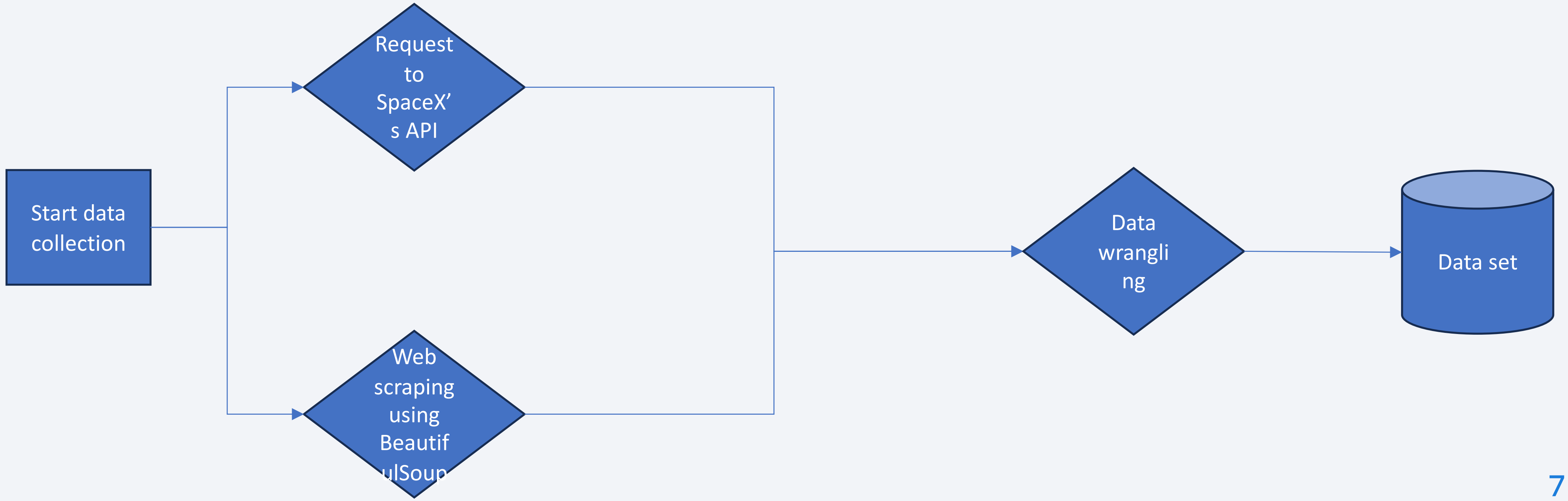
Methodology

Executive Summary

- Data collection methodology:
  - Request to the SpaceX API + Web scraping using BeautifulSoup
- Perform data wrangling
  - Dealing with missing values + changing data type for landing outcomes + creating landing outcome label from Outcome column
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Standardizing data + splitting data (train + test) + finding the best parameter (GridSearchCV) + Testing algorithms

Data Collection

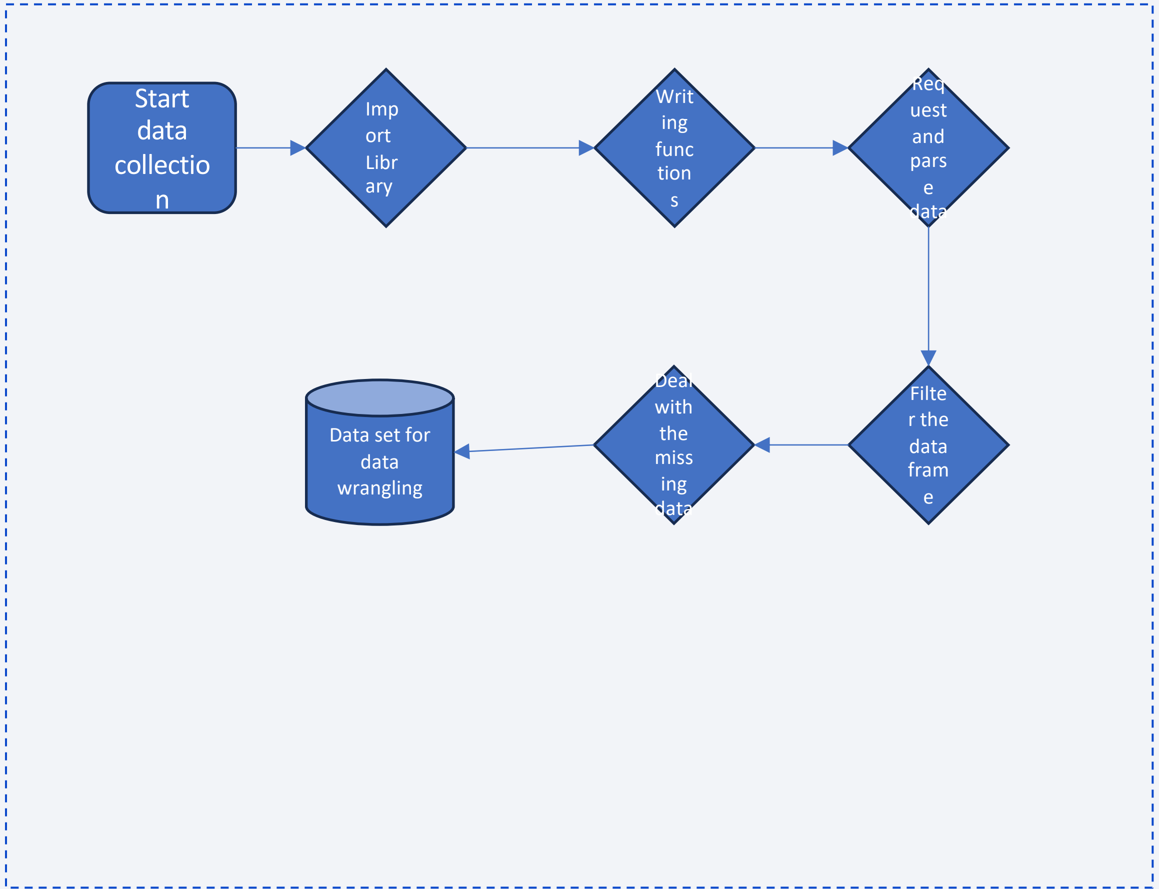
- We collect data using SpaceX's API and web scraping and then clean the data.



Data Collection – SpaceX API

- Start data collection
- Import library
- Writing collection functions
- Request and parse data
- Filter the dataframe
- Deal with the missing data

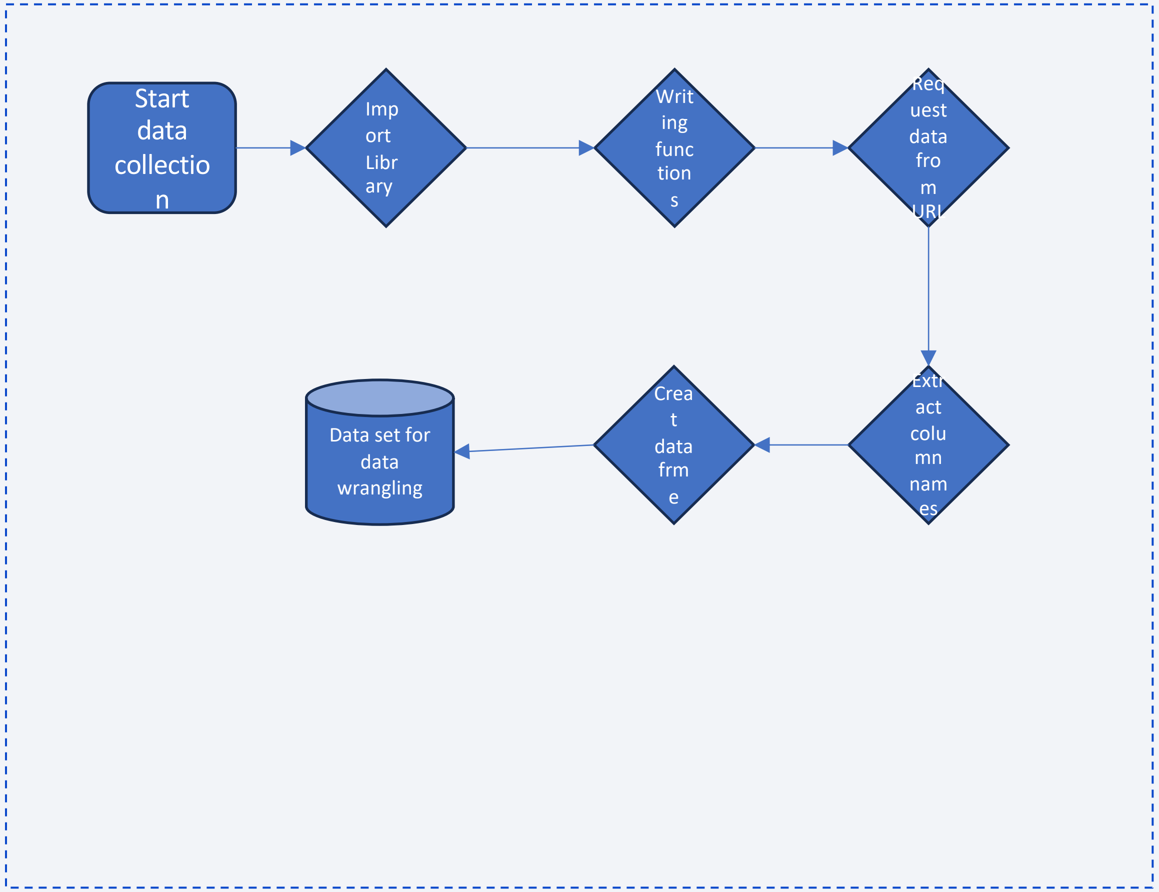
- https://github.com/weiw1422/WeiSpace/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Start data collection
- Import library
- Writing collection functions
- Request data from the URL
- Extract column names
- Create data frame

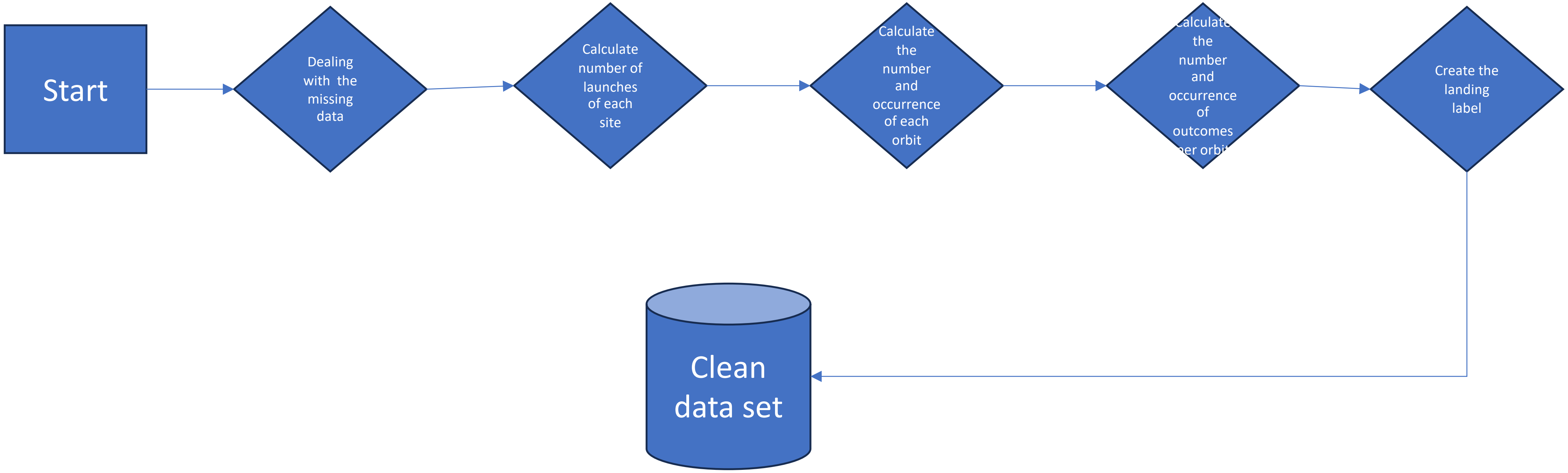
- https://github.com/weiw1422/WeiSpace/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- Start data wrangling
- Dealing with the missing data
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcome per orbit type
- Create a landing outcome label from Outcome column
- https://github.com/weiw1422/WeiSpace/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

Data Wrangling

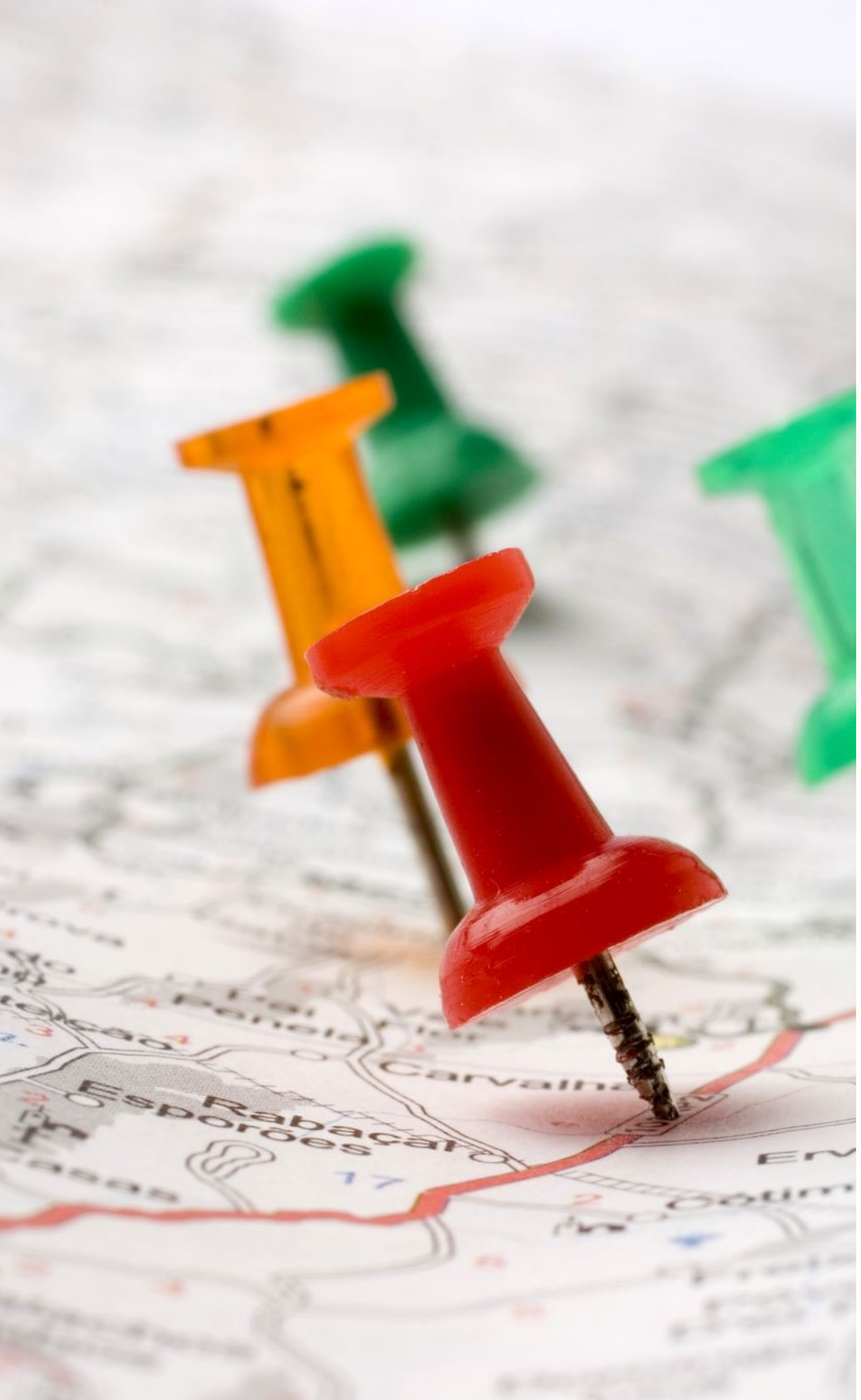


EDA with Data Visualization

- Charts plotted (The aim of these charts is to find the features relevant to the success of launches):
- Flight number vs. payload mass (hue = ‘Class’, scatter)
- Flight number vs. launch site (hue = ‘Class’, scatter)
- Launch site vs. payload mass (hue = ‘Class’, scatter)
- Success rate of each orbit (bar chart)
- Flight number vs. orbit (hue = ‘Class’, scatter)
- Payload mass vs. orbit (hue = ‘Class’, scatter)
- Success rate vs. date (line)
- https://github.com/weiw1422/WeiSpace/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(1).ipynb

EDA with SQL

- The SQL queries performed
- %sql select DISTINCT "Launch_Site" from SPACEXTBL
- %sql select * from SPACEXTBL where "Launch_Site" like 'KSC%' LIMIT 5
- %sql select SUM(PAYLOAD_MASS__KG_) as SUM_PAYLOAD_MASS from SPACEXTBL where "Customer" = 'NASA (CRS)’
- %sql select AVG(PAYLOAD_MASS__KG_) as AVG_PAYLOAD_MASS from SPACEXTBL where "Booster_Version" = 'F9 v1.1’
- %sql select MIN("Date") as min_date from SPACEXTBL where "Landing_Outcome" = 'Success (drone ship)’
- %sql select "Booster_Version" from SPACEXTBL where "Landing_Outcome" = 'Success (ground pad)' and "PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_" < 6000
- %sql select "Mission_Outcome", count("Mission_Outcome") as "Total number of successful and failure mission outcomes" from SPACEXTBL group by "Mission_Outcome”
- %sql select DISTINCT "Booster_Version" from SPACEXTBL where "PAYLOAD_MASS__KG_" = (select MAX("PAYLOAD_MASS__KG_") from SPACEXTBL)
- %sql select substr(Date, 4, 2) as "Month name", "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTBL where substr(Date,7,4)='2017' and "Landing_Outcome" = 'Success (ground pad)’
- %sql select "Landing_Outcome", COUNT("Landing_Outcome") as "Successful Landing Outcome" from SPACEXTBL where "Date" between '04/06/2010' and '20/03/2017' and "Landing_Outcome" like 'Success%' group by "Landing_Outcome" ORDER by "Successful Landing Outcome" DESC
- https://github.com/weiw1422/WeiSpace/blob/main/jupyter-labs-eda-sql-edx_sqllite%20(1).ipynb



Build an Interactive Map with Folium

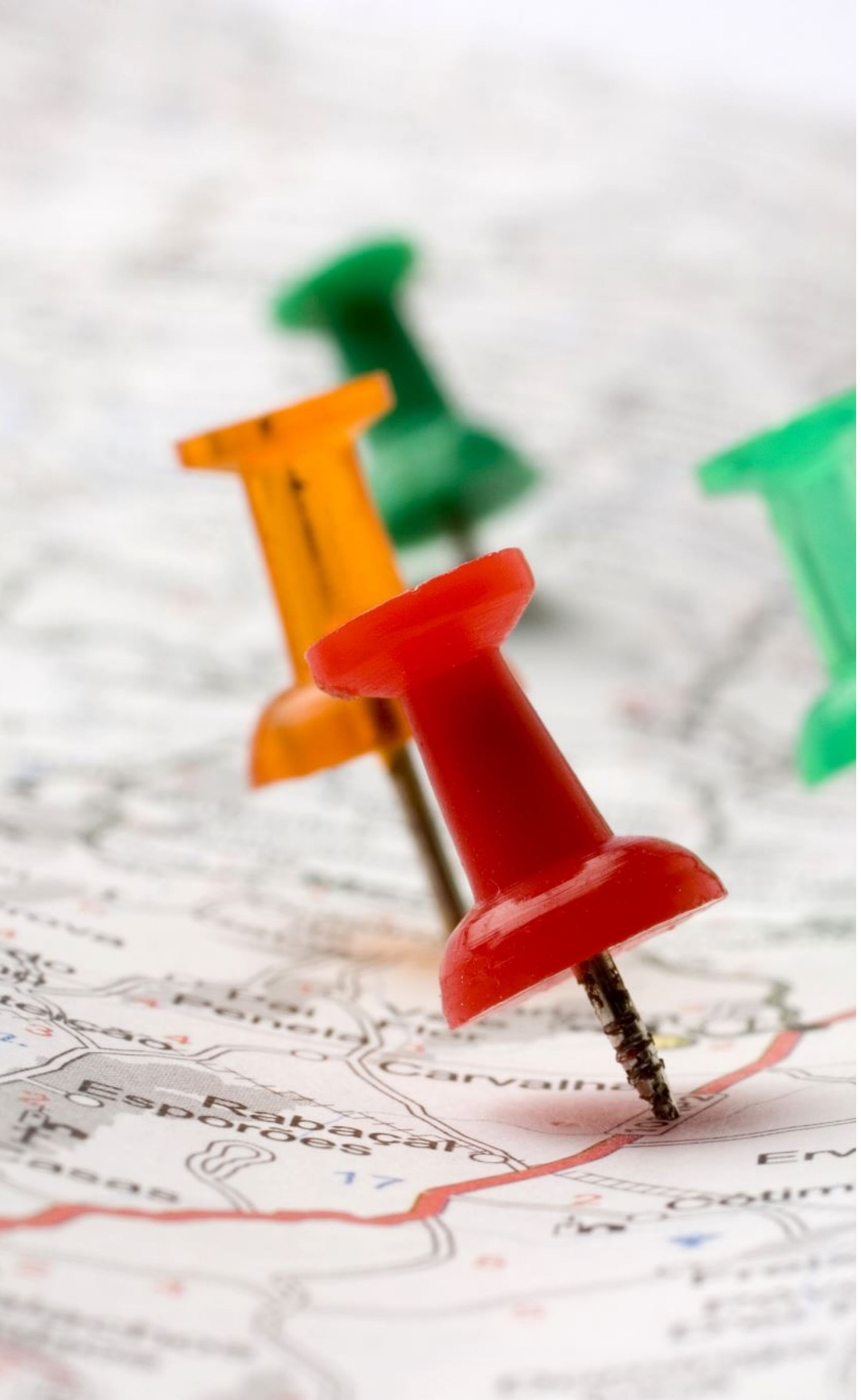
```
circle = folium.Circle([lat,lon], radius=1000, color='#d35400', fill=True).add_child(folium.Popup(site))

 marker = folium.map.Marker(

    [lat,lon],

    # Create an icon as a text label

    icon=DivIcon(

        icon_size=(20,20),

        icon_anchor=(0,0),

        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % site,

        )

    )
```

Aim: To mark each launch site.

Build an Interactive Map with Folium

- marker_cluster = MarkerCluster()
- site_map.add_child(marker_cluster)

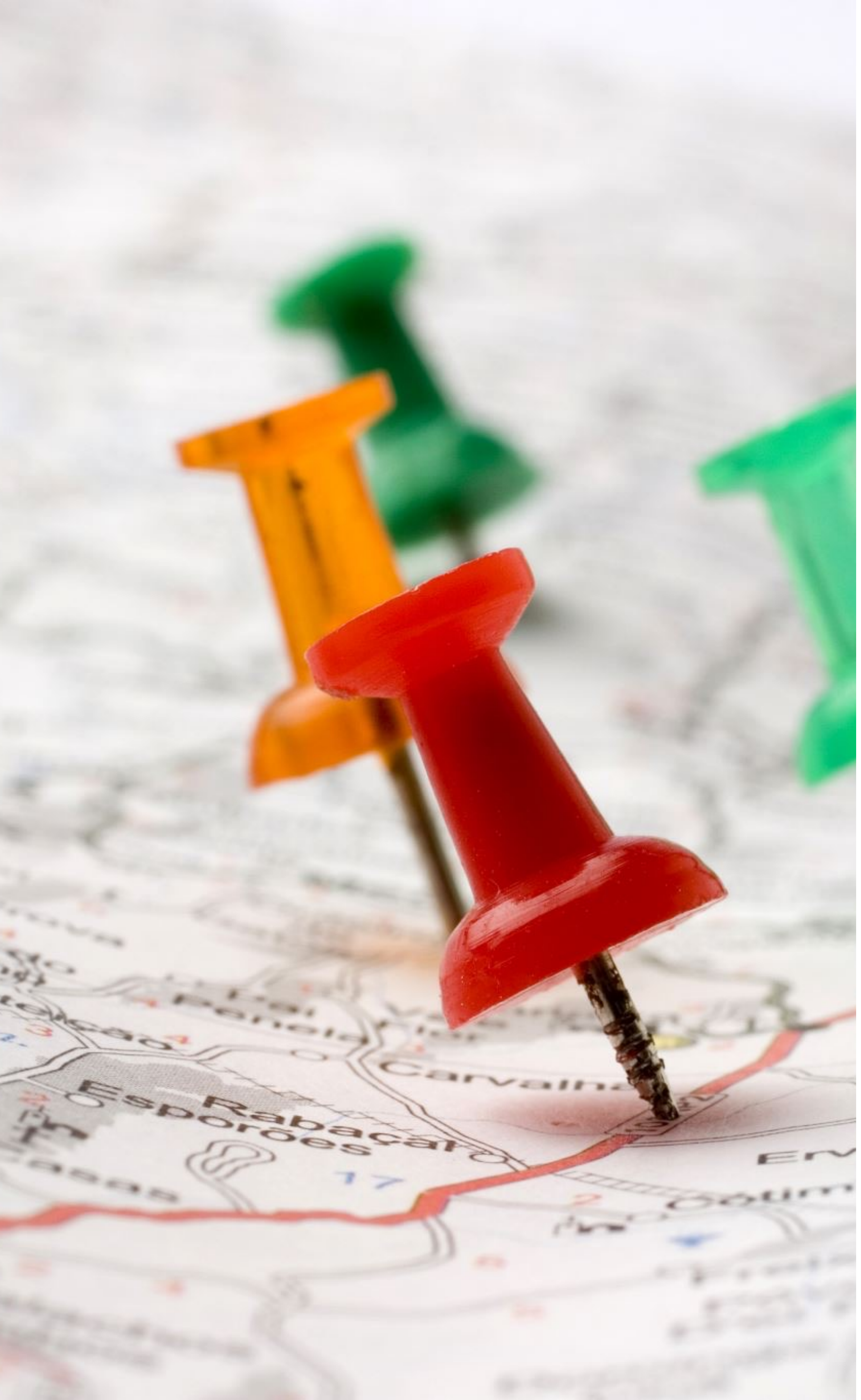
- # for each row in spacex_df data frame
- # create a Marker object with its coordinate
- # and customize the Marker's icon property to indicate if this launch was successed or failed,
- # e.g., icon=folium.Icon(color='white', icon_color=row['marker_color']
- for index, record in spacex_df.iterrows():
-     lat = record['Lat']
-     lng = record['Long']
-     label = record['Launch Site']
-     # TODO: Create and add a Marker cluster to the site map
-     marker = folium.Marker(
-         location=[lat, lng],
-         icon=folium.Icon(color = record['maker_color']),
-         popup=label,
-     )
-     # marker = folium.Marker(...)
-     marker_cluster.add_child(marker)
- Aim: Create markers for all launch sites.



Build an Interactive Map with Folium

```
formatter = "function(num) {return L.Util.formatNum(num, 5);};"
mouse_position = MousePosition(
position='topright',
separator=' Long: ',
empty_string='NaN',
lng_first=False,
num_digits=20,
prefix='Lat:',
lat_formatter=formatter,
lng_formatter=formatter,
)
```

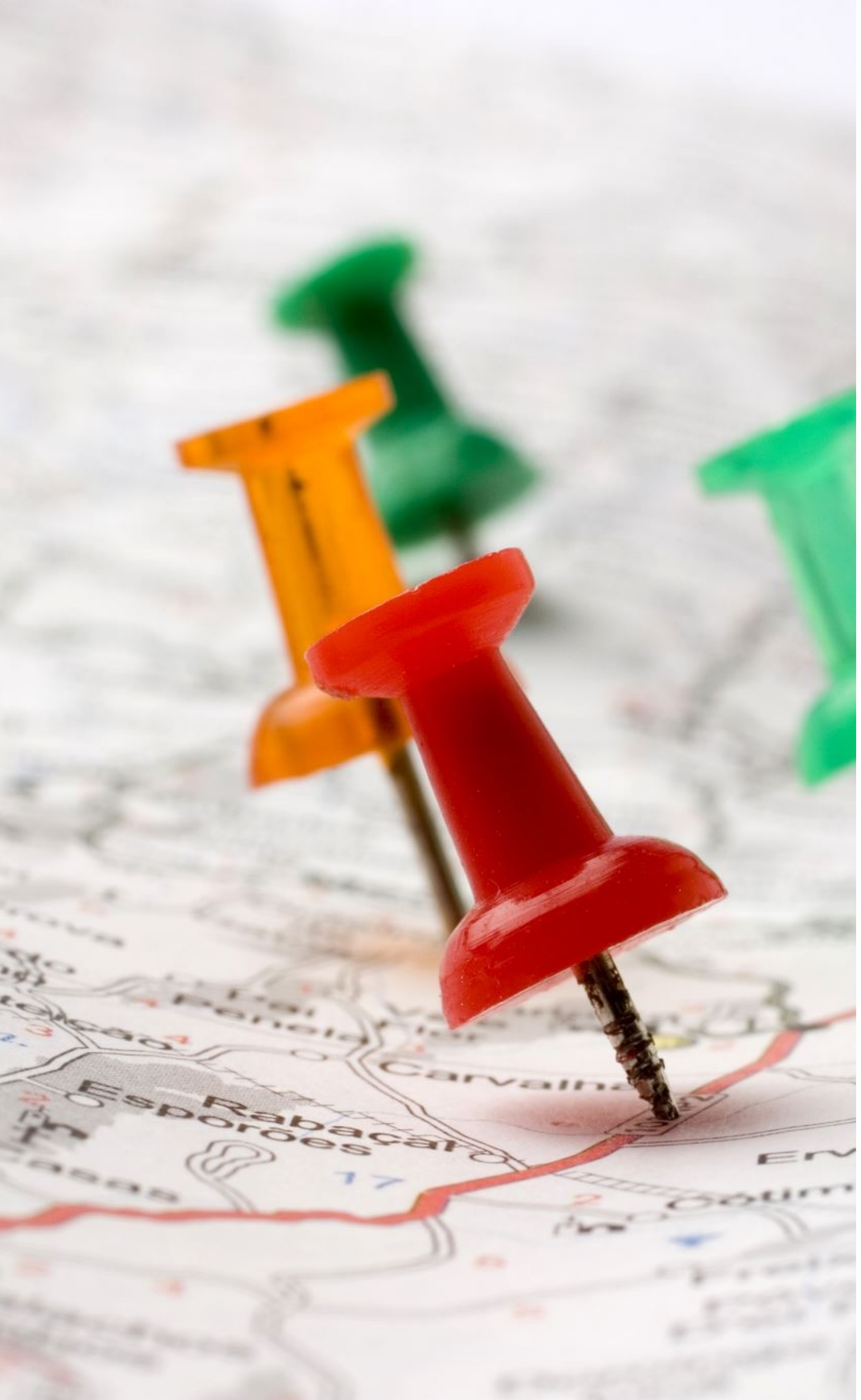
Aim: Add Mouse Position to get the coordinate (Lat, Long) for a mouse over on the map



Build an Interactive Map with Folium

```
distance_marker = folium.Marker(
   [coastline_lat,coastline_lon],
   icon=DivIcon(
       icon_size=(20,20),
       icon_anchor=(0,0),
       html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % "{:10.2f} KM".format(distance_coastline),
       )
    #popup='closed coastline',
   )

site_map.add_child(distance_marker)
```

Aim: Create and add a folium.Marker on your selected closest coastline point on the map and display the distance between coastline point and launch site using the icon property



Build an Interactive Map with Folium

```
lines=folium.PolyLine(locations=[[launch_site_lat,launch_site_lon],[coastline_lat,coastline_lon]], weight=1)

site_map.add_child(lines)
```

Aim: Create a `folium.PolyLine` object using the coastline coordinates and launch site coordinate.

```
site_map.add_child(distance_marker_city)
lines_city=folium.PolyLine(locations=[[launch_site_lat,launch_site_lon],[city_lat,ci
site_map.add_child(lines_city)


highway_lat = 28.56224
highway_lon = -80.57062
distance_highway = calculate_distance(launch_site_lat, launch_site_lon, highway_lat,
distance_marker_highway = folium.Marker(
   [highway_lat,highway_lon],
   icon=DivIcon(
       icon_size=(20,20),
       icon_anchor=(0,0),
       html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % "{:10.2f}
       )
    #popup='closed coastline',
   )

site_map.add_child(distance_marker_highway)
lines_highway=folium.PolyLine(locations=[[launch_site_lat,launch_site_lon],[highway_
site_map.add_child(lines_highway)
```

Build an Interactive Map with Folium

- Aim: Create a marker with distance to a closest highway and draw a line between the marker to the launch site. The same method is also used to create a closest railway and city, which we omit here.

Build an Interactive Map with Folium

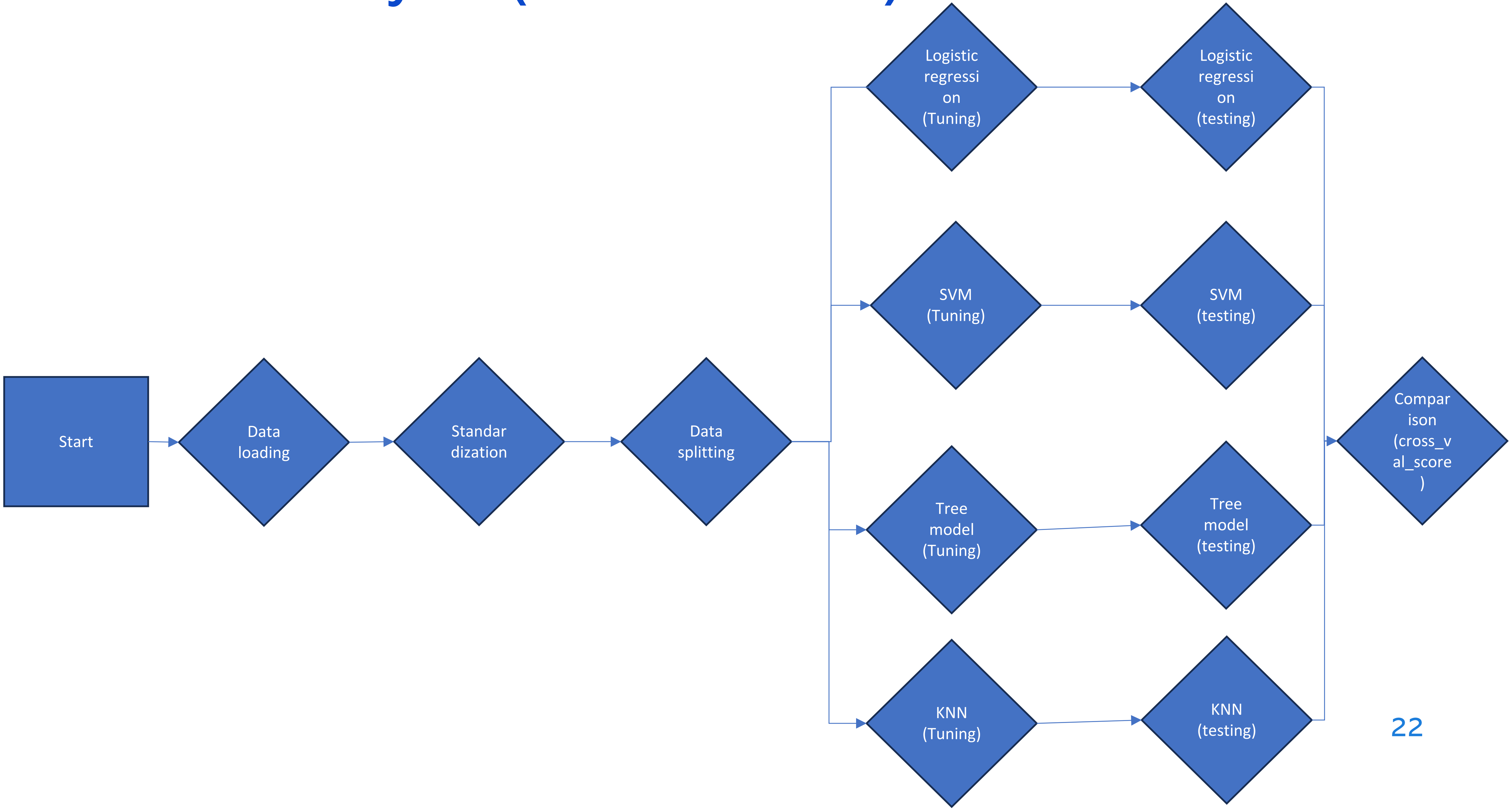
- https://github.com/weiw1422/WeiSpace/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb
- Maps:
- https://github.com/weiw1422/WeiSpace/blob/main/map1.png
- https://github.com/weiw1422/WeiSpace/blob/main/map2.1.png
- https://github.com/weiw1422/WeiSpace/blob/main/map2.2.png
- https://github.com/weiw1422/WeiSpace/blob/main/map2.3.png
- https://github.com/weiw1422/WeiSpace/blob/main/map2.4.png
- https://github.com/weiw1422/WeiSpace/blob/main/map3.png

Build a Dashboard with Plotly Dash

- Launch site dropdown, including:
- A pie chart demonstrating the success counts four all sites.
- Four pie charts showing the success rates for each launch site.
- A scatter plot with a slider range bar, illustrating the relationship between payload and mission outcomes.
- Using the Plotly Dash, we would like to find solutions to the following questions:
- Which site has the largest successful launches?
- Which site has the highest launch success rate?
- Which payload range(s) has the highest launch success rate?
- Which payload range(s) has the lowest launch success rate?
- Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate?

https://github.com/weiw1422/WeiSpace/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)



Predictive Analysis (Classification)

- The process of the predictive analysis is as follows.
- Data loading
- Data standardization
- Data splitting
- Algorithm tuning for logistic regression, SVM, tree model and KNN (GridSearchCv)
- Test the above four algorithms
- Comparison between the four algorithms using cross_val_score (cv = 10)

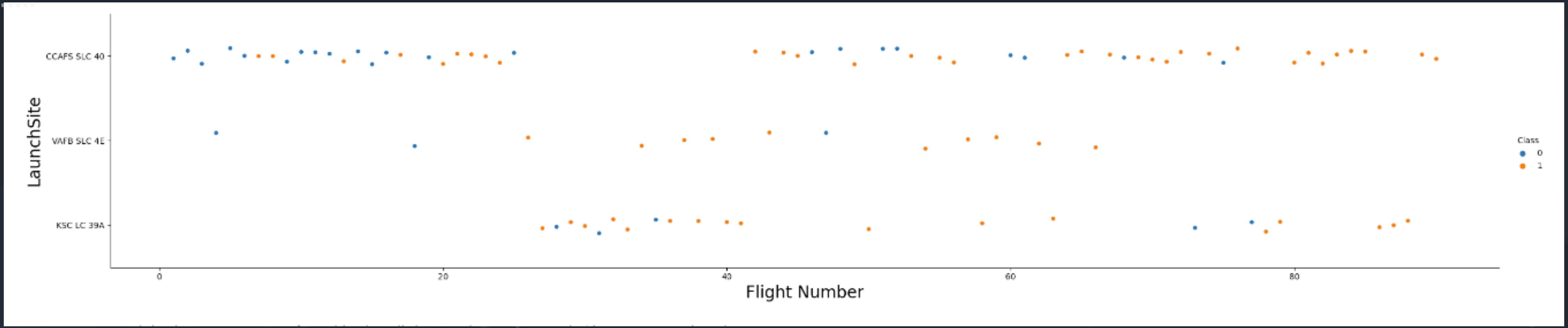
- https://github.com/weiw1422/WeiSpace/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(2).ipynb

Results

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

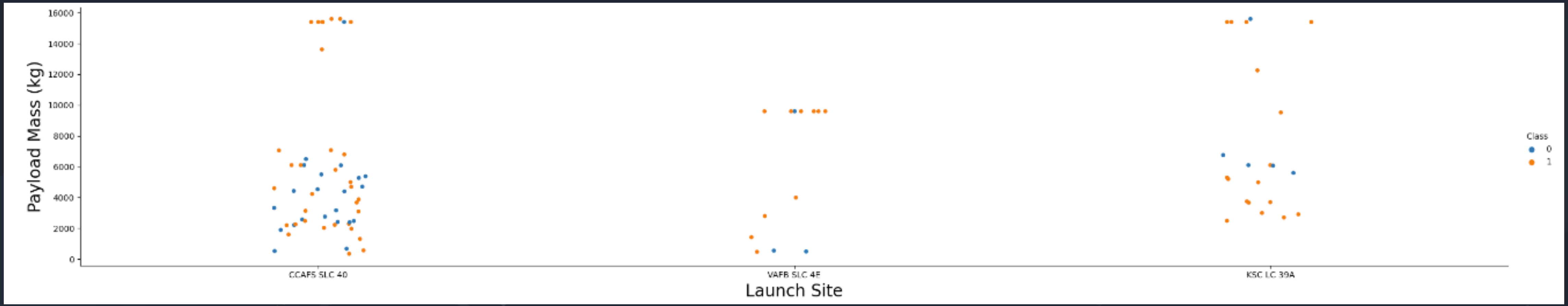
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

- CCAFS SLC-40 enjoys the highest number of flight number, followed by KSC LC-39A. VAFB SLC-4E launches the least number of flights.

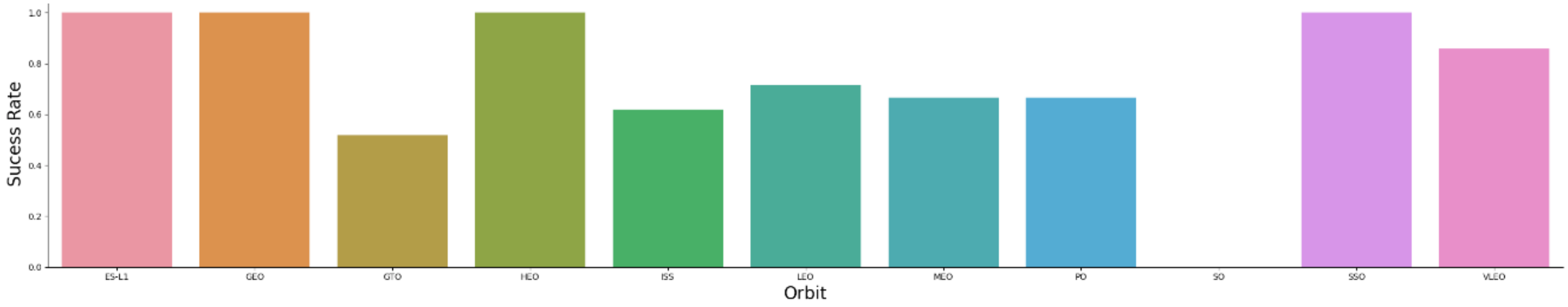


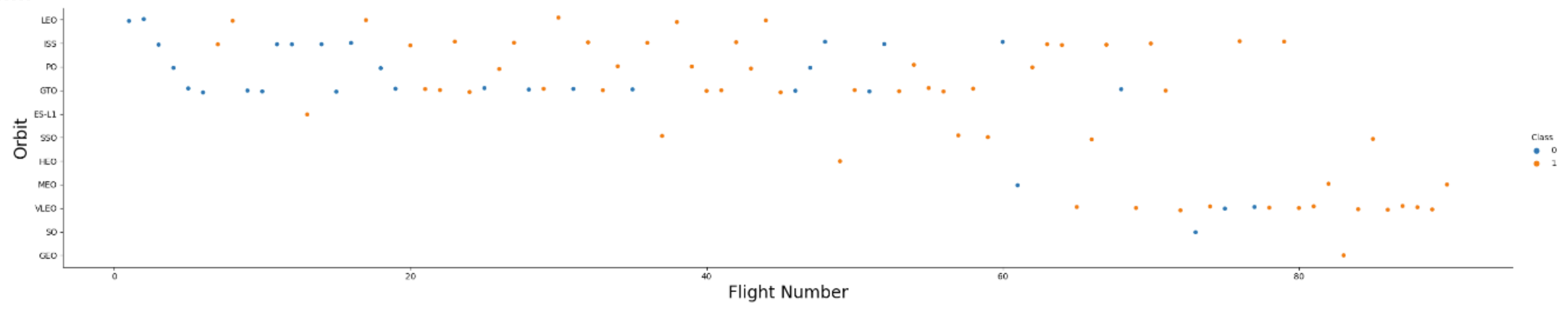
Payload vs. Launch Site

- There are no rockets launched for heavy payload mass (greater than10000) from VAFB SLC-4E.

Success Rate vs. Orbit Type

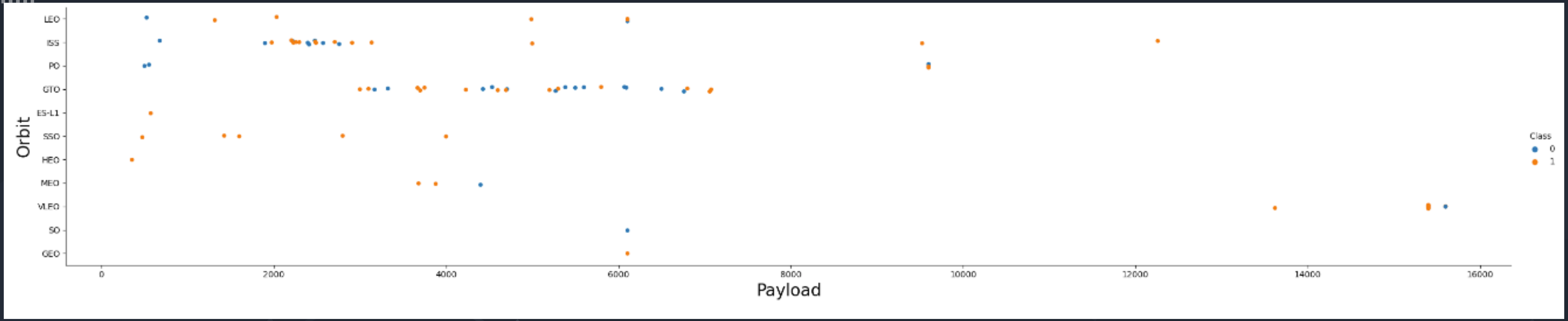
- ES-L1, GEO, HEO and SSO enjoy the highest success rate, while SO has 0 success rate.





Flight Number vs. Orbit Type

- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

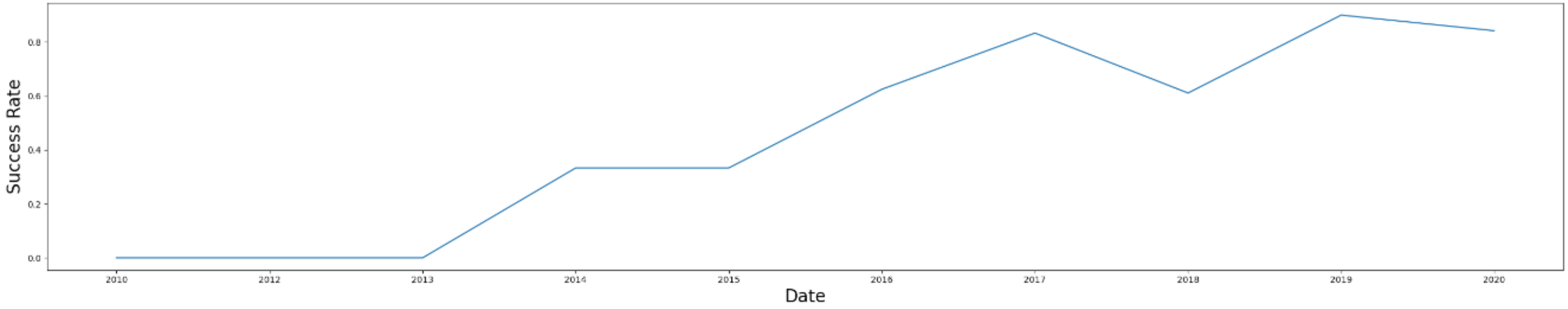


Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However, for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

- The success rate since 2013 kept increasing till 2017.



All Launch Site Names

- The right panel shows the names of the unique launch sites. There are 4 launch sites in total: CCAFS LC-40, VAFB SLC-4E, KSC LC-39A and CCAFS SLC-40.
- **%sql** select DISTINCT "Launch_Site" from SPACEXTBL
- %sql is the magic function embedded in Jupiter notebook, which executes sql queries. Distinct gets the unique names. “Launch_Site” is the column in the SPACEXTBL table. The keywork select is to display information.

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

Launch Site Names Begin with 'KSC'

- The following table demonstrates the launch site names starting with ‘KSC’.
- **%sql** select * from SPACEXTBL where "Launch_Site" like 'KSC%' LIMIT 5
- * means all rows, ``where’’ shows the condition,  ``like + %’’ extract the string pattern starting with ‘KSC’.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
19/02/2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490.0	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
16/03/2017	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600.0	GTO	EchoStar	Success	No attempt
30/03/2017	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300.0	GTO	SES	Success	Success (drone ship)
05/01/2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300.0	LEO	NRO	Success	Success (ground pad)
15/05/2017	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070.0	GTO	Inmarsat	Success	No attempt

Total Payload Mass

Calculate the total payload carried by boosters from NASA

45596.0

The total payload carried by boosters from NASA is 45596.0.

%sql select SUM(PAYLOAD_MASS__KG_) as SUM_PAYLOAD_MASS from SPACEXTBL where "Customer" = 'NASA (CRS)’

SUM is the build-in function that calculate the sum of the values in a column.

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is shown as follows: AVG_PAYLOAD_MASS: 2928.4.
- %sql select AVG(PAYLOAD_MASS__KG_) as AVG_PAYLOAD_MASS from SPACEXTBL where "Booster_Version" = 'F9 v1.1’
- AVG is the build-in function that calculates the average of the values in a column which satisfies the condition after ``where’’.

First Successful Ground Landing Date

min_date
04/08/2016

The first successful landing outcome on drone ship happened on 04/08/2016.

```
%sql select MIN("Date") as min_date from SPACEXTBL
where "Landing_Outcome" = 'Success (drone ship)'
```

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 are displayed on the right panel.
- %sql select "Booster_Version" from SPACEXTBL where "Landing_Outcome" = 'Success (ground pad)' and "PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_" < 6000

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Mission_Outcome	Total number of successful and failure mission outcomes
None	0
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- The total number of successful and failure mission outcomes is shown in the above table, which gives that most of missions are accomplished.
- %sql select "Mission_Outcome", count("Mission_Outcome") as "Total number of successful and failure mission outcomes" from SPACEXTBL group by "Mission_Outcome"

Boosters Carried Maximum Payload

- The right panel list the names of the booster which have carried the maximum payload mass. All are the F9 B5 type carriers.
- %sql select DISTINCT "Booster_Version" from SPACEXTBL where "PAYLOAD_MASS__KG_" = (select MAX("PAYLOAD_MASS__KG_") from SPACEXTBL)

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

2017 Launch Records

- The following table lists the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

Month name	Landing_Outcome	Booster_Version	Launch_Site
02	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
01	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
03	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
07	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

- %sql select substr(Date, 4, 2) as "Month name", "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTBL where substr(Date,7,4)='2017' and "Landing_Outcome" = 'Success (ground pad)'

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

- The right table ranks the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

- %sql select "Landing_Outcome", COUNT("Landing_Outcome") as "Successful Landing Outcome" from SPACEXTBL where "Date" between '04/06/2010' and '20/03/2017' and "Landing_Outcome" like 'Success%' group by "Landing_Outcome" ORDER by "Successful Landing Outcome" DESC

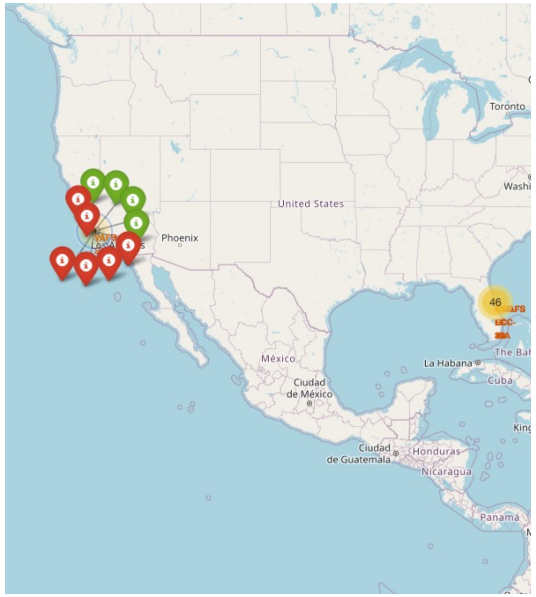
Landing_Outcome	Successful Landing Outcome
Success	20
Success (drone ship)	8
Success (ground pad)	7

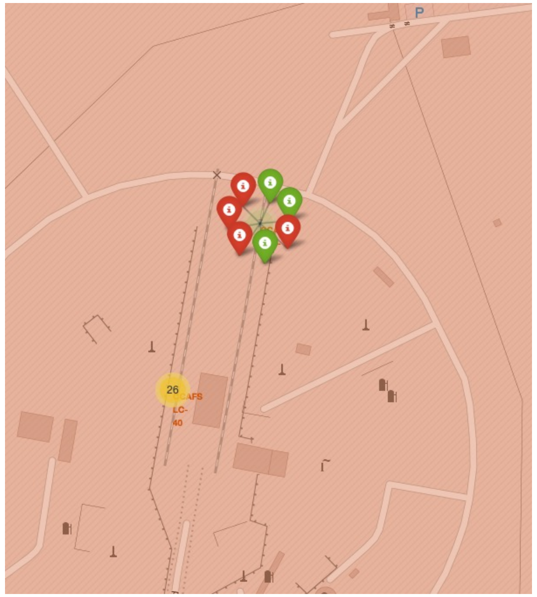
Section 3

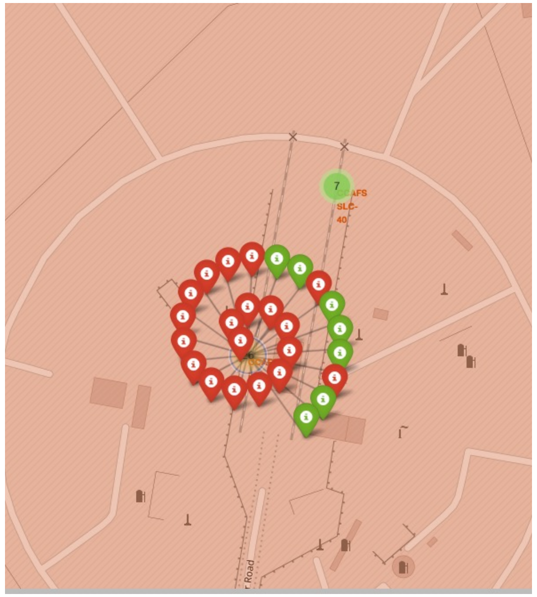
Launch Sites Proximities Analysis

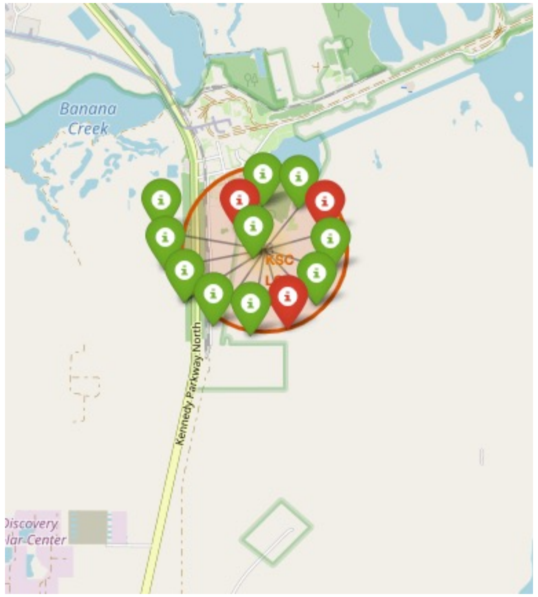
Locations of launch sites

- There are 4 launch sites across the US. While VAFB SLC 4E is located in California, the other three (CCAFS SLC-40, CCAFS LC-40 and KSC LC-29A) sit in Florida.







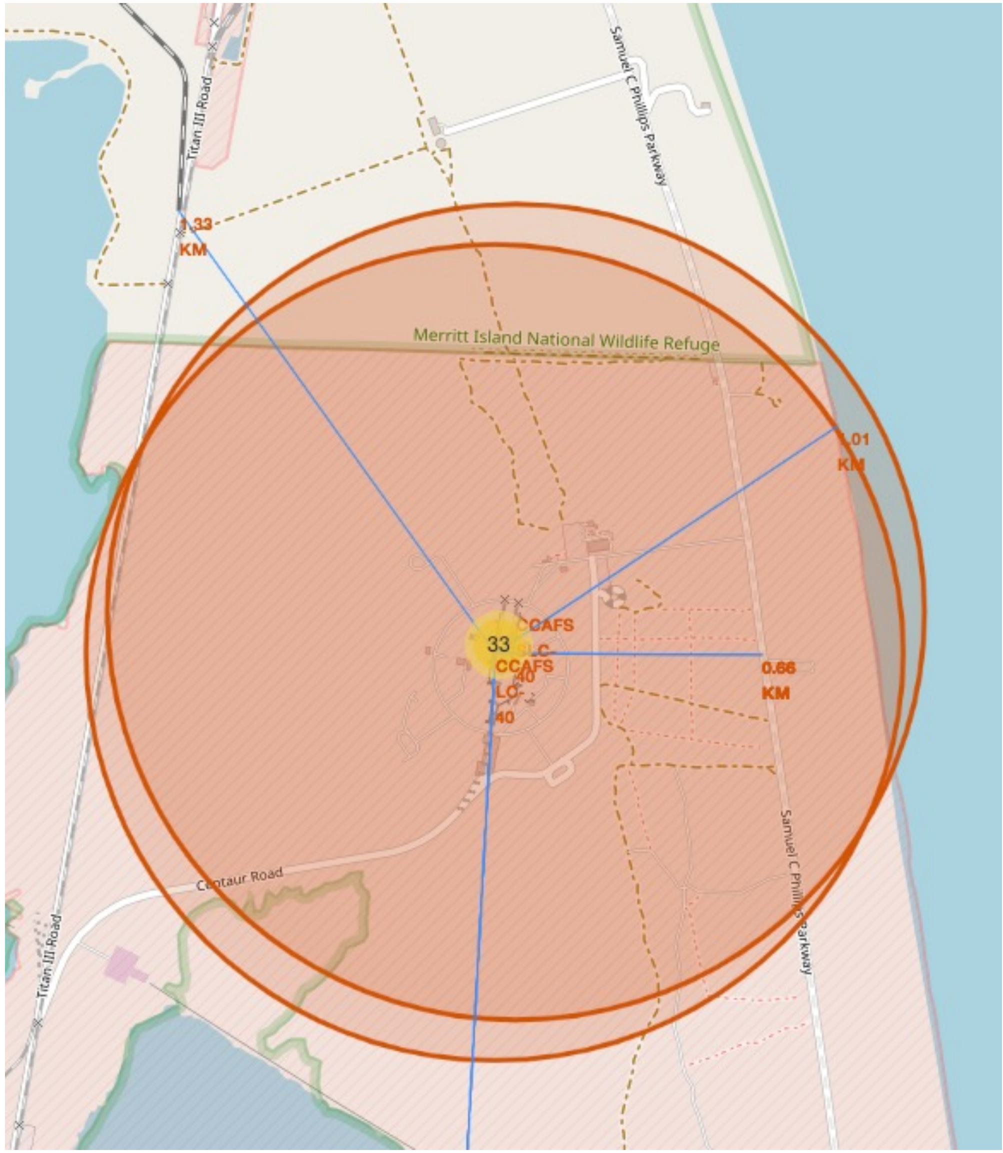


Launch outcomes for each site

- The first map shows the launch outcomes for VAFB SLC-4E. There are 10 rockets launched from this site, with 4 success launches and 6 failures.
- The second map shows the launch outcomes for CCAFS SLC-40. There are 7 rockets launched from this site, with 3 success launches and 4 failures.
- The third map shows the launch outcomes for CCAFS LC-40. There are 26 launches, with 7 successes and 19 failures
- The last map shows the launch outcomes for KSC LC-39A. There are 13 launches, with 10 successes and 3 failures.

Launch site CCAFS LC-40 to its proximities

- The right panel shows the distance from launch site CCAFS LC-40 to railway (1.33km), highway (0.66km) and coastline (1.01km) respectively. It can be seen from the graph that the launch site is closed to those traffic lines so that then equipment and material for the rocket launch can be delivered economically.

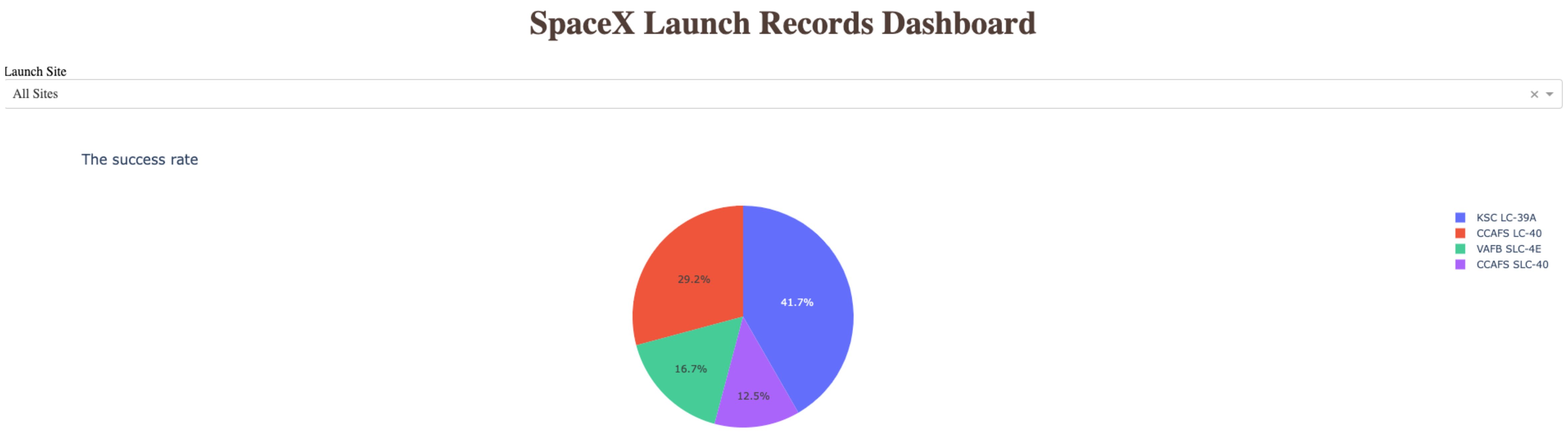


Section 4

Build a Dashboard with Plotly Dash

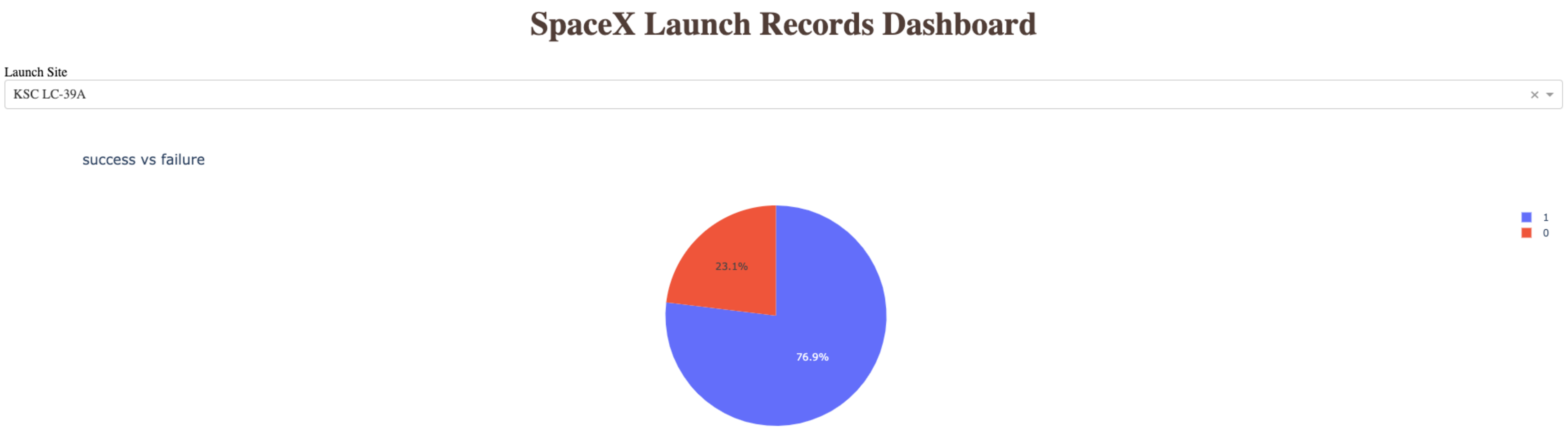
Launch success rates for all sites

- KSC LC-39A enjoys the highest success rate, while CCAFS SLC-40 has the least success rate.



Launch site with highest launch success ratio

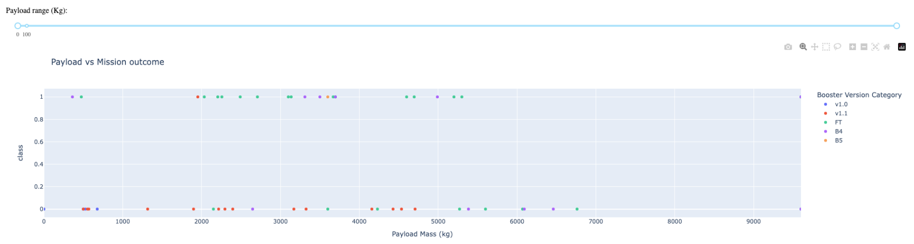
- For KSC LC-39A, 76.9% launch is successful, and the rest of 23.1% launch fails.

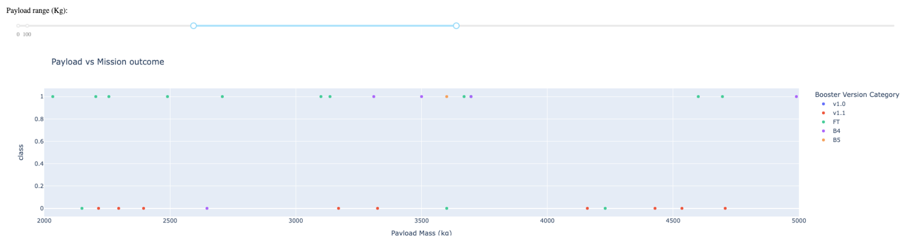


Payload vs. Launch Outcome

- The first panel demonstrates the relationship between payload and launch outcomes for various booster versions. Ft performs best, followed by B4. V1.0 and B5 have zero success rates.

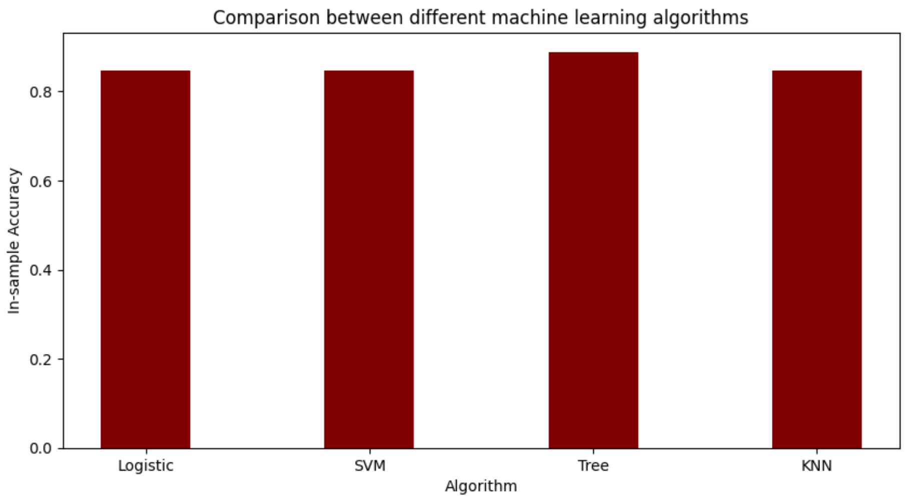
- The payload range for the second graph is from 2000 to 5000. It is easy to see that only FT has been launched successfully in the interval [2000,3200].

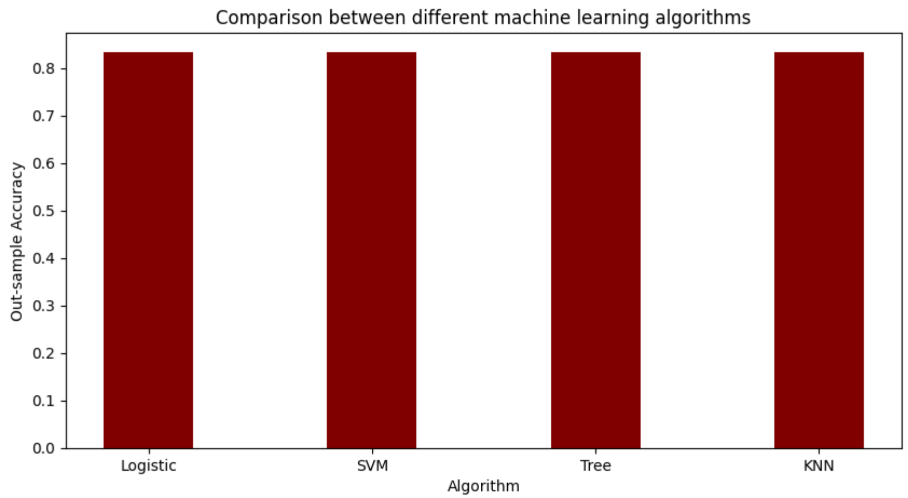




Section 5

Predictive Analysis (Classification)



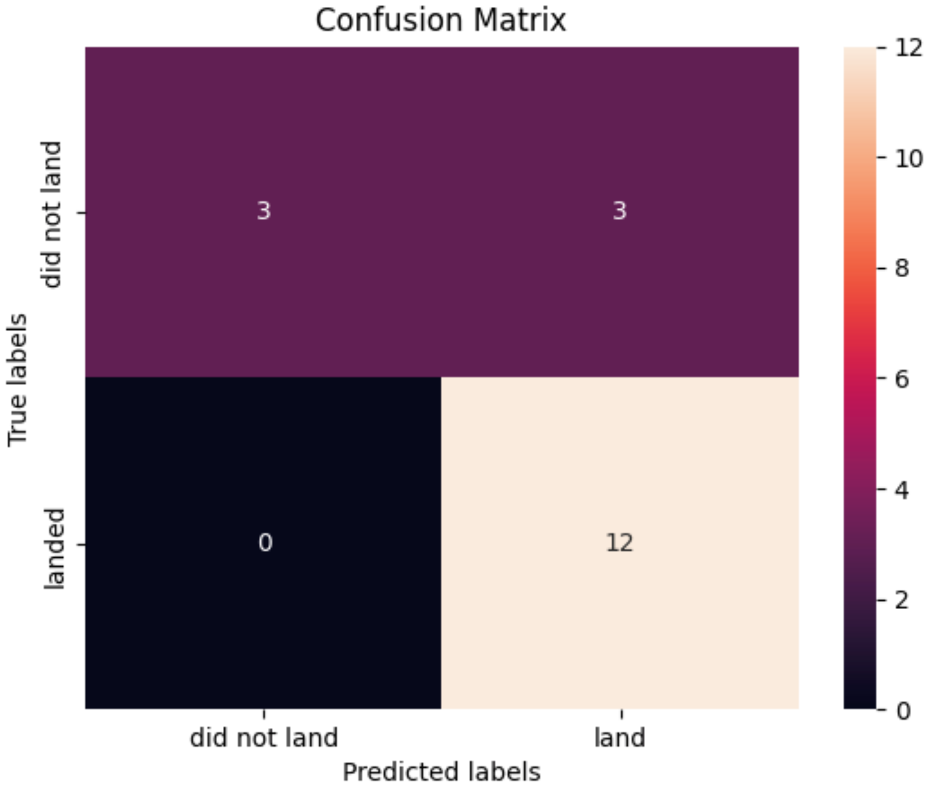


Classification Accuracy

- The left panel shows the in-sample accuracy of the four machine learning algorithms used in the assignment. The tree model enjoys the highest in-sample accuracy with 0.8875
- The right panel shows the out-sample accuracy of the four machine learning algorithms. The accuracy of the four algorithms is the same, with value being 0.8333333333333334.

Confusion Matrix

- As the out-sample accuracy is almost the same for the four algorithms, I use the cross_val_score to measure the accuracy of the algorithms. (cv = 10). The mean scores of the four algorithms are listed as follows. Logistic regression: 0.81, SVM: 0.8, Tree: 0.8, KNN: 0.8. Therefore, the best model is Logistic regression for this case.



Conclusions

- The features relevant to successful launch and landing would be orbits, reused count, block, payload mass, serial, GridFins and launch sites.
- KSC LC-39A enjoys the highest successful rate and thus would be the optimal launch site.
- A desirable launch site should be closed to highways, railways, and coastlines.
- The best machine learning method to predict successful landings would be the logistic regression model.

Appendix

All Jupiter notebooks can be found as follows.

Data collection (API and webscraping)

https://github.com/weiw1422/WeiSpace/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

https://github.com/weiw1422/WeiSpace/blob/main/jupyter-labs-webscraping.ipynb

Data wrangling

https://github.com/weiw1422/WeiSpace/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA (visualization and sql)

https://github.com/weiw1422/WeiSpace/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(1).ipynb

https://github.com/weiw1422/WeiSpace/blob/main/jupyter-labs-eda-sql-edx_sqllite%20(1).ipynb

Interactive visualization (folium and plotly dash)

https://github.com/weiw1422/WeiSpace/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

https://github.com/weiw1422/WeiSpace/blob/main/spacex_dash_app.py

Machine learning predictive analysis

https://github.com/weiw1422/WeiSpace/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(2).ipynb

Thank you!