IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tianxiang Yang
2024-10-05

Outline

- Executive Summary
- Introduction
- Methodology
- Results

  Insights drawn from EDA; Launch Sites Proximities Analysis; Build a Dashboard with Plotly Dash; Predictive Analysis (Classification)

- Conclusion
- Appendix

Executive Summary

Summary of methodologies

- Gain the useful data: by data collection methodology and data wrangling
- Tell narrative stories: by exploratory data analysis (EDA) using visualization and SQL
- Disclose additional information: by interactive visual analytics using Folium and Plotly Dash
- Formulate quantitative relationships: by predictive analysis using classification models

Summary of all results

- Based on current data, which of them are the key factors for making success launch?
- Besides superficial data, what may also affect the launch outcome and other aspect of business development?
- Can launch outcome be predicted at a high-level accuracy using current data science?

Introduction

- The commercial space age is here, companies are making space travel affordable for everyone. Among the most successful explorer is SpaceX, whose endeavors on first-stage rocket is both attractive and instructive. The broad and deep insights of their current data, especially that drawn officially, will provide lots of informative instructions to empower the industry players.

- In this ever changing world, data is mushrooming in huge volume, variety and velocity, no exception for commercial space industries and their program. Here, thanks to SpaceX APIs and the webpages, we will be pleased to discuss the factors that influence the rocket launches programs through data-writing stories.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Gather data from the SpaceX REST API endpoints, calling the .json() method
  - Scarp tables from web pages of Wikipedia, using Python BeautifulSoup package
- Perform data wrangling
  - Present data attributes, replace missing values and transfer data for further training 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
  - Perform exploratory analysis by training labels and find best models and hyperparameters

Data Collection

- 1) Request to the SpaceX REST API;
- 2) Clean the requested data;
- 3) Extract a Falcon 9 launch records HTML table from Wikipedia;
- 4) Parse the table and convert it into a Pandas data frame

Gather data from the SpaceX REST API endpoints, calling the .json() method

JSON on the Space X REST API → **DATAFRAME of pandas** → **Optimized DATA-FRAME of pandas**

Scarp tables from web pages of Wikipedia, using Python BeautifulSoup package

HTML webpage on Wikipedia → **HTML TABLE on the webpages** → **Optimized DATA-FRAME of pandas**

Data Collection – SpaceX API

- GitHub URL of the completed SpaceX API calls notebook: https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb, as an external reference and peer-review purpose

Gather data from the SpaceX REST API endpoints, calling the .json() method

JSON on the Space X REST API → **DATAFRAME of pandas** → **Optimized DATA-FRAME of pandas**

```
spacex_url = 'https://api.spacexdata.com/v4/launches/past'
response = requests.get(spacex_url)
re2JSON = response.json()
data = pd.json_normalize(re2JSON)
```

Data Collection - Scraping

- GitHub URL of the completed web scraping notebook: https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb, as an external reference and peer-review purpose

Scarp tables from web pages of Wikipedia, using Python BeautifulSoup package

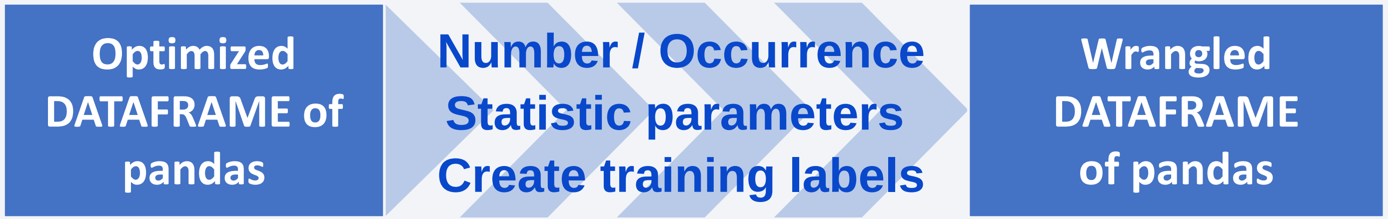
HTML webpage on Wikipedia → **TABLE on the HTML webpages** → **Optimized DATA-FRAME of pandas**

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).text
soup = BeautifulSoup(response)
html_tables = soup.find_all('table’)
Extract DATAFRAME messages from TABLE based on TAG tree

Data Wrangling

- 1) Perform some exploratory data analysis for meaningful information
- 2) Determine training labels by deriving data for further process

Wrangle data collected from the SpaceX REST API endpoints, calling different dataframe- and list- method



- GitHub URL of completed data wrangling related notebooks: https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data wrangling.ipynb, as an external reference and peer-review purpose

EDA with Data Visualization

- 1) Perform some exploratory data analysis for meaningful information
- 2) Prepare and perform Data Feature Engineering for meaningful purposes

Visualize data collected and wrangled from the SpaceX REST API endpoints, using matplotlib/seaborn package

Wrangled DATAFRAME of pandas → **SCATTER plot: for relationship of 2 variables**
BAR plot: for data comparison among groups
LINE plot: for trend of 1 variable with another

- GitHub URL of my completed EDA with data visualization notebook : https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/edadataviz.ipynb, as an external reference and peer-review purpose

EDA with SQL

- **select distinct <field> from SPACEXTABLE** : list the <field> values
- **… from SPACEXTABLE where <field> = _** : judge the <field> condition fully
- **… from SPACEXTABLE where <field> like '_%'** : judge the <field> condition partially
- **… from SPACEXTABLE where <num field> between _ and _** : judge the <num field> condition based on range
- **… where … limit <num>** : display the first <num> records of the result
- **select sum/min(<field>) from SPACEXTABLE** : present summed/minimum data of <field>
- **select count(*) from SPACEXTABLE** : present total count of records
- **… round(<num field>, <N>)** … : return <num field> data to that in <N> decimal places
- **… from SPACEXTABLE group by <field>** : summarize data when <field> is grouped
- **… from SPACEXTABLE order by <field> (desc)** : present data in ascending (descending) order of <field>
- **… substr(<date field>,6,2)/substr(<date field>,6,2)** … : return <data field> data as month/year

- GitHub URL of my completed EDA with SQL notebook: https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb, as an external reference and peer-review purpose

Build an Interactive Map with Folium

- **Background** : folium.Map(location=<coor>, zoom_start=_) to display the basic map figure as background
- **Circle** : folium.Circle(<coor>, radius=_, color='_', fill=True) to display point shapes such as launch centers
- **Popup** : folium.Popup('_') to display the label (even onto other objects)
- **MapMarker** : folium.map.Marker(<coor>, icon=DivIcon()) to display annotations such as centers' attributes
- **MarkerCluster** : MarkerCluster() with .add_child method to display overlapped objects properly such as annotations of launch results at single site
- **MousePosition** : MousePosition(position='_', separator='_', empty_string='NaN', lng_first=False, num_digits=_, prefix='_', lat_formatter=formatter, lng_formatter=formatter,) to display mouse attributes
- **Marker** : folium.Marker(<coor>, icon=DivIcon()) to display annotations such as the distance
- **Polyline** : folium.Polyline([<coor1>,<coor2>], weight=_) to display polylines such as linkages between the test point and launch sites

- GitHub URL of my completed interactive map with Folium map : https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

- **SUCCESS-PIE-CHART (interaction: launch sites)**

When the input is 'ALL', it will present the total success launches among sites, to disclose on which site the largest successful launches may occur; When the input is a specific site name, it will present the proportion of successful launches of that single site, to disclose on which site the highest launch success ratios occur

- **SUCCESS-PAYLOAD-SCATTER-CHART (interaction: launch sites, payload range)**

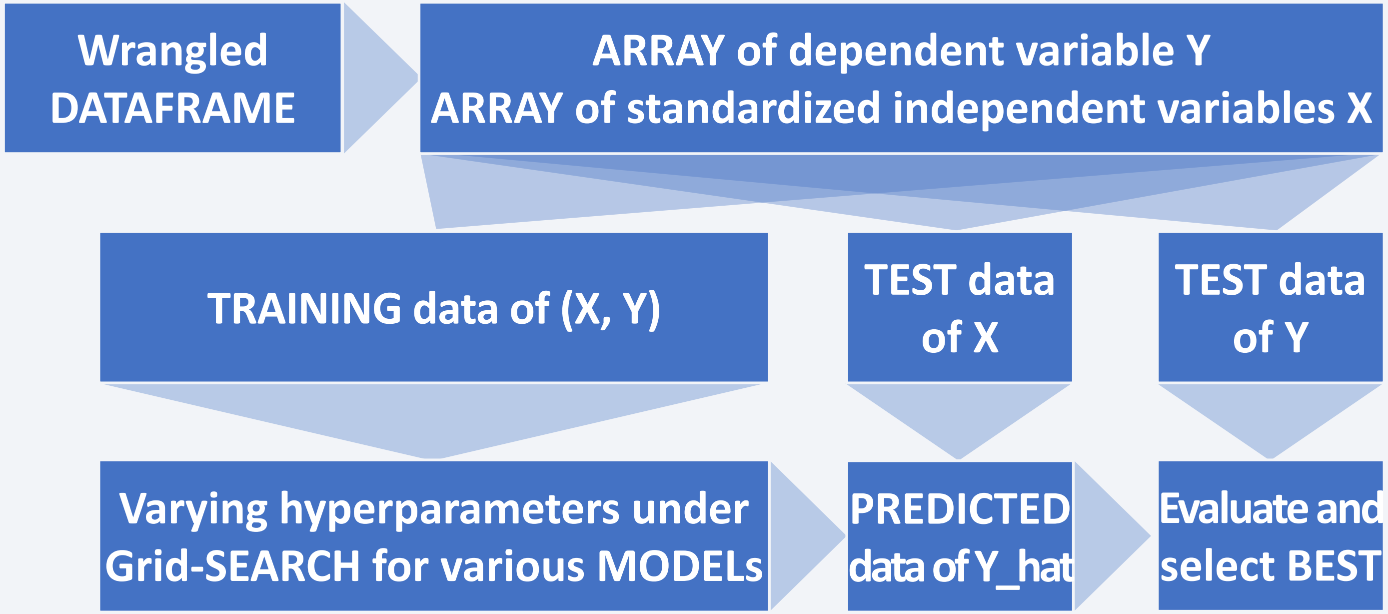
When input1 is 'ALL', it will use all cases; When input1 is a specific site name, it will use the site-specific cases. When input2 is set at a payload range, it will use the cases whose Payload Mass (kg) matches that range. For the predetermined cases, it will present how successful launches correlate to Payload Mass (kg) and Booster Version Category, to disclose in which payload range, by which booster version and on which site the highest launch success ratios occur

- GitHub URL of my completed Plotly Dash lab: https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- 1) Create a column of the dependent variable Y
- 2) Standardize the data of standardized independent variables X whichever
- 3) Split records of (X, Y) into training and test data
- 4) Find the methods fitting best using test data

Perform exploratory analysis by training labels and find best models and hyperparameters



- GitHub URL of my completed predictive analysis lab: https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine Learning Prediction_Part_5.ipynb, as an external reference and peer-review purpose

Results

- Exploratory data analysis results

a) Flight Num positively correlates with launch success ratios at site CCAFS SLC 40
b) Payload Mass (kg) correlates weakly with launch success ratios in general
c) Launch Site correlates weakly with launch success ratios in general
d) Orbit correlates with launch success ratios to a certain extent in general
e) Year positively correlates with launch success ratios in general

- Interactive analytics demo in screenshots

a) Please check and download my Github source file: https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py
b) Launch the Cloud IDE and follow the instructions, the demo will be running on http://127.0.0.1:8050

- Predictive analysis results

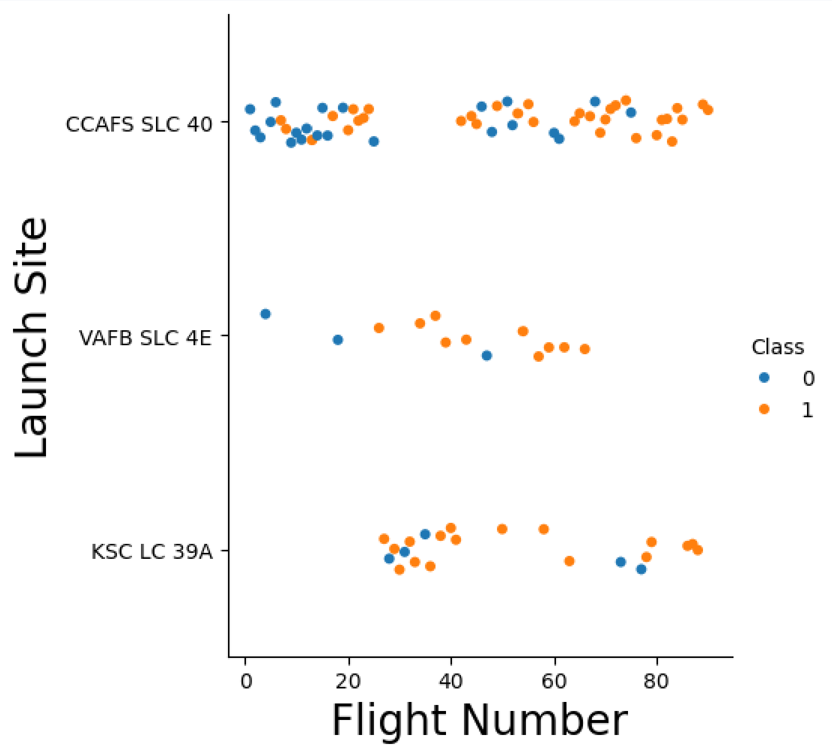
a) The decision tree classifier built on below hpyerparameters {'criterion': 'entropy', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'} fits best so far
b) Its accuracy on test data of (X, Y) is almost 0.90

Section 2

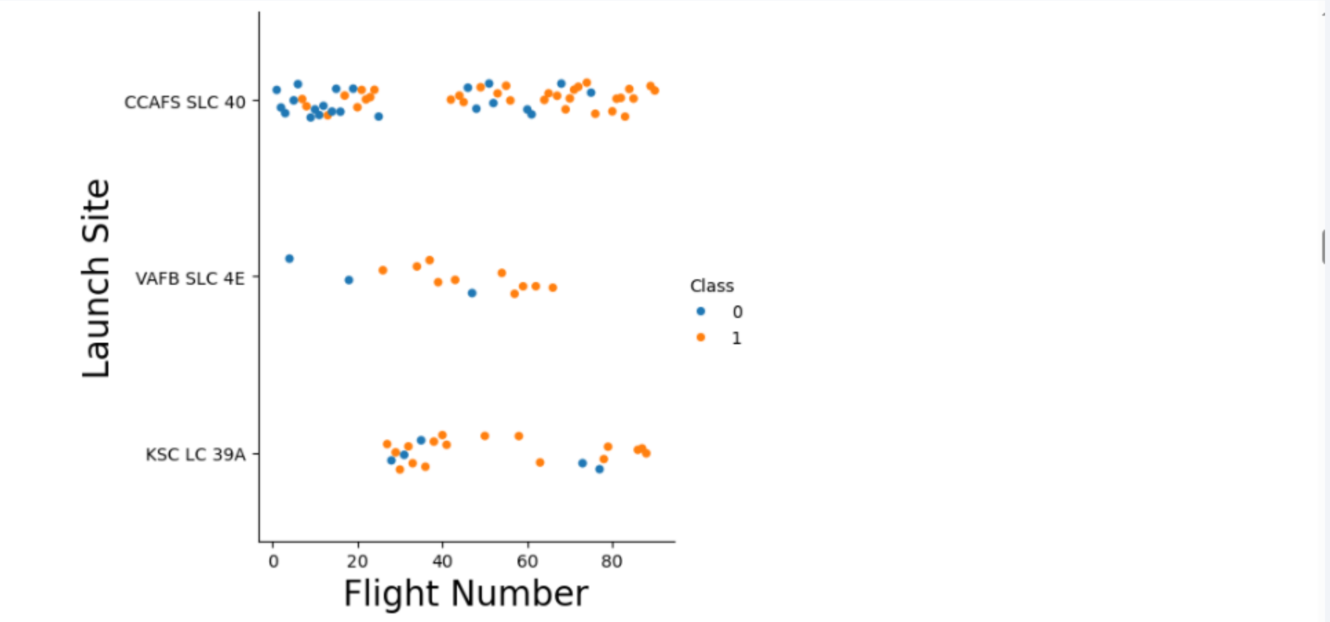
Insights drawn from EDA

Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site



- Screenshot of the scatter plot with explanations

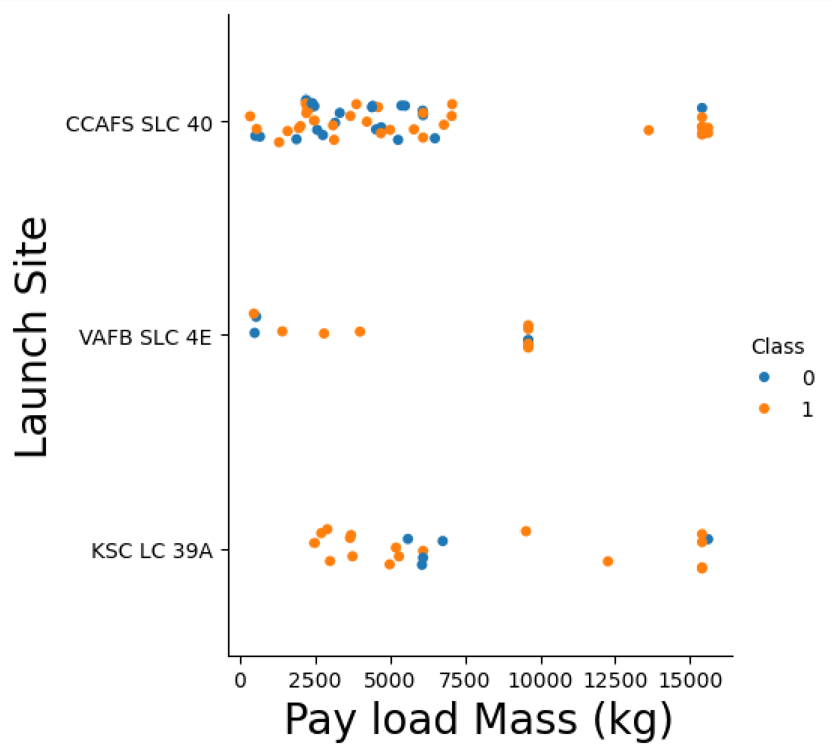



```
[*]: # Flight Number positively correlates with rate of success launch at site CCAFS SLC 40
```

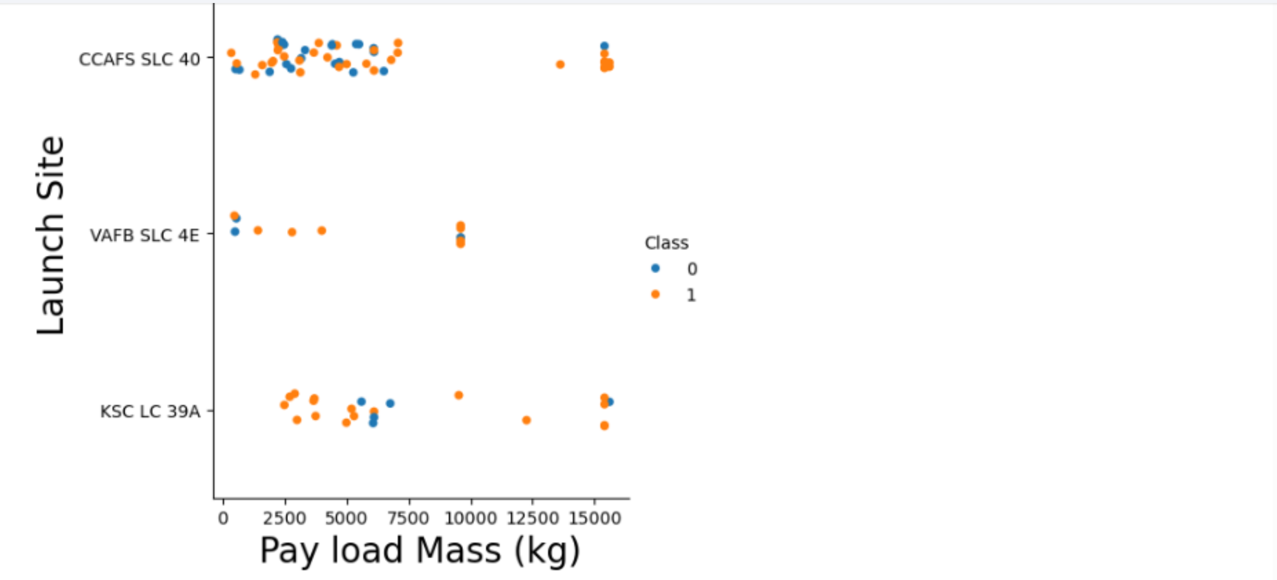
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site



- Screenshot of the scatter plot with explanations

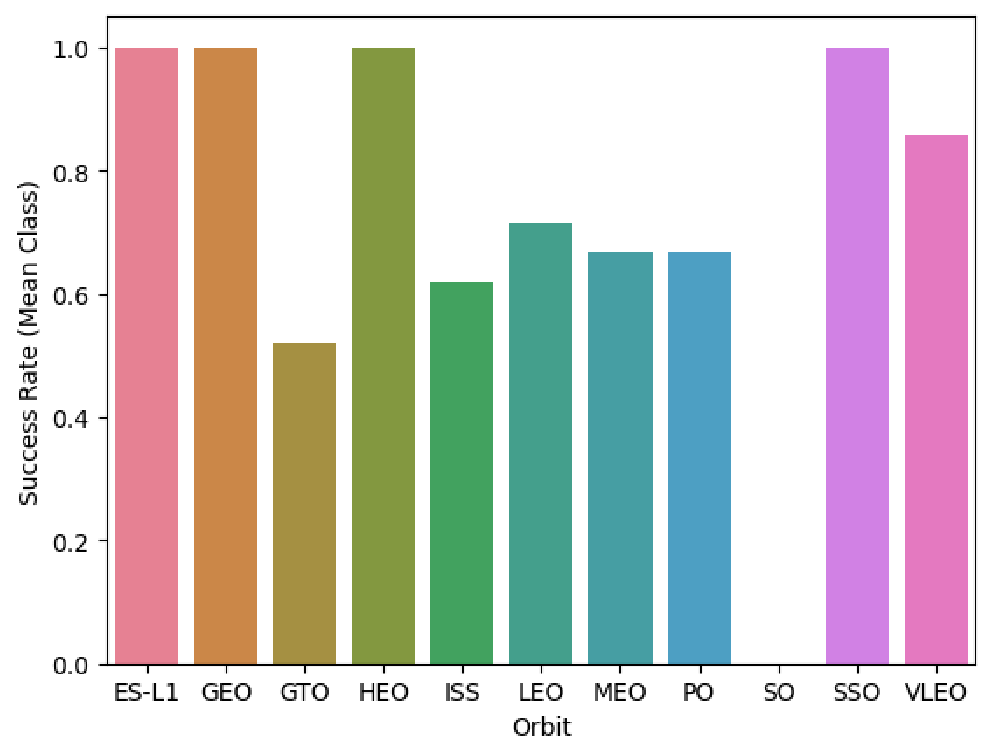



```
# Pay load Mass (kg) correlates weakly with rate of success launch in general
# Launch Site correlates weakly with rate of success launch in general
# More launches in site CCAFS SLC 40 and less in site VAFB SLC 4E
```

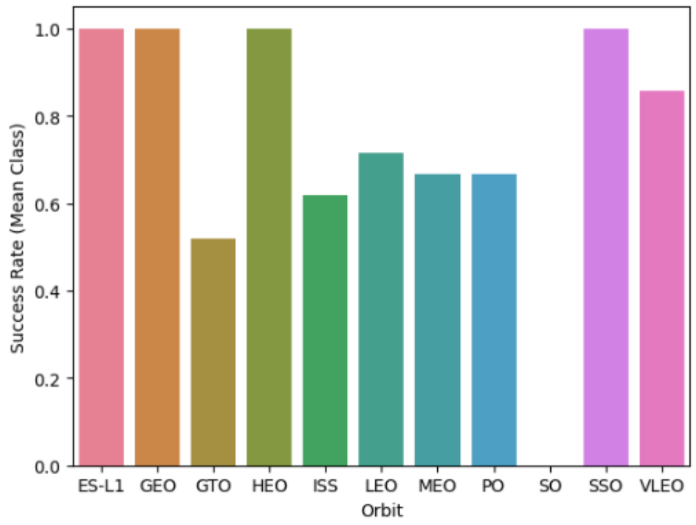
Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type
- Screenshot of the scatter plot with explanations



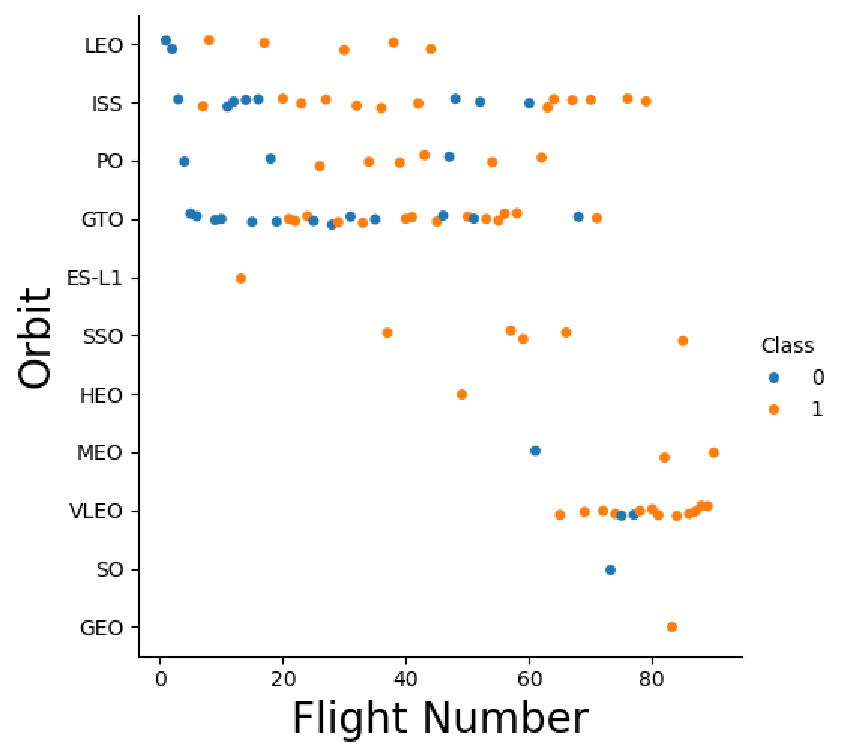

```
[22]: <AxesSubplot:xlabel='Orbit', ylabel='Success Rate (Mean Class)'>
```

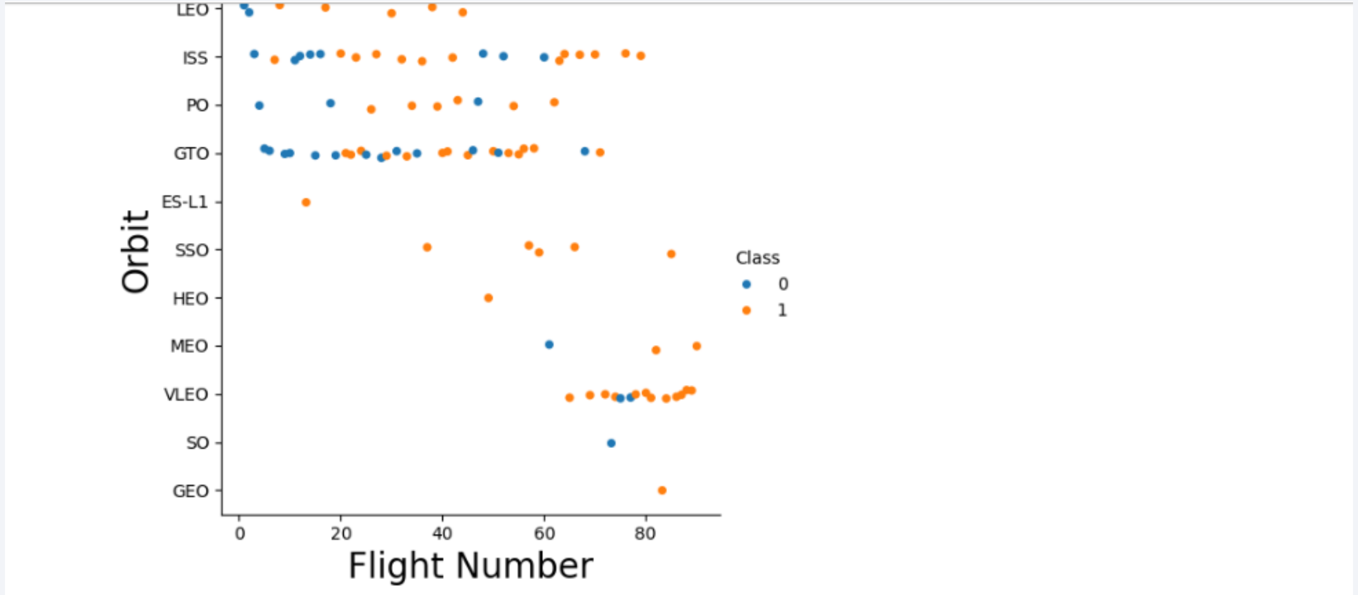



```
[ ]: # Orbit correlates with rate of success launch to a certain extent in general
     # Orbit ES-L1, GEO, HEO and SSO get the highest (full) rate of success launch
     # Orbit SO get the lowest (zero) rate of success launch
```

Flight Number vs. Orbit Type

- Scatter point of Flight number vs. Orbit type
- Screenshot of the scatter plot with explanations



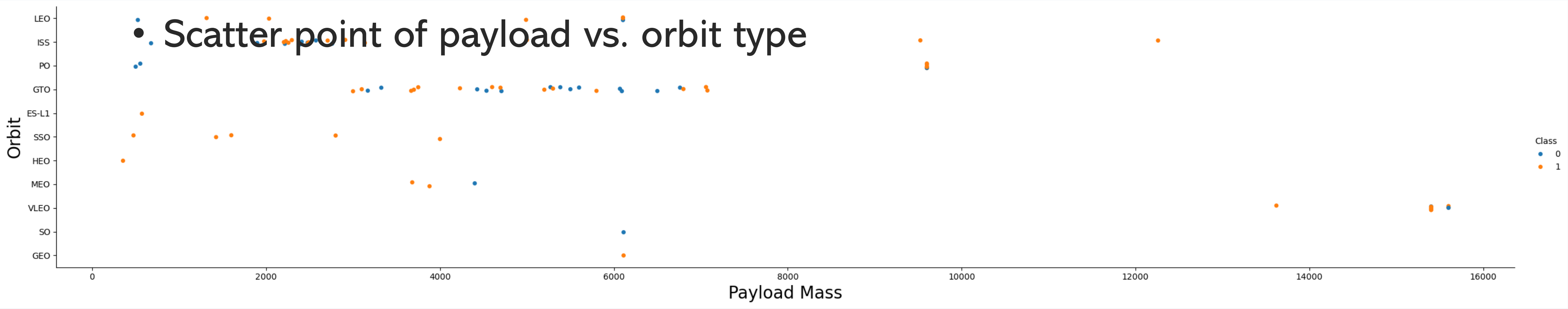



```
[ ]: # Flight Number correlates weakly with rate of success launch in general
     # Orbit correlates with rate of success launch to a certain extent in general
```

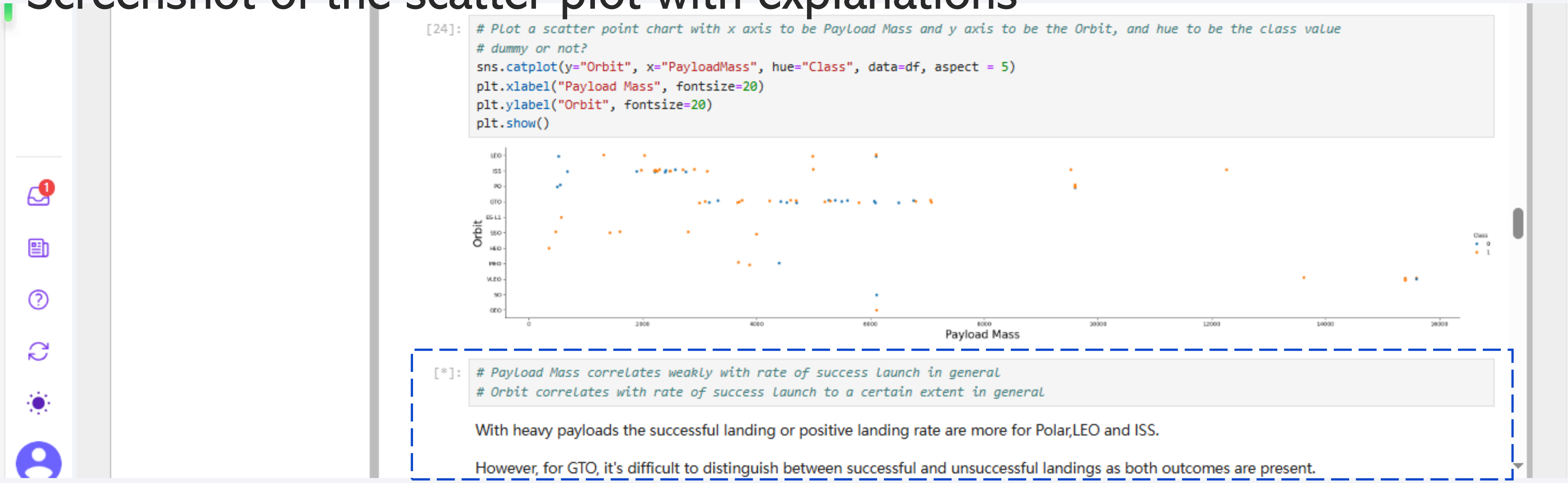
You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

Payload vs. Orbit Type

- Scatter point of payload vs. orbit type

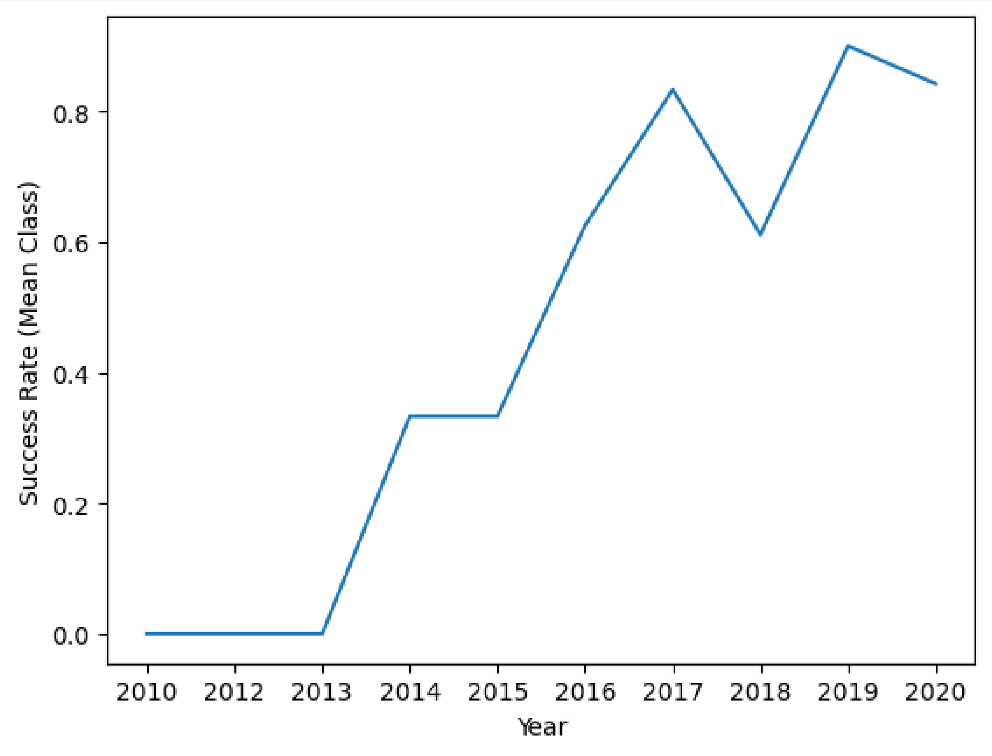


- Screenshot of the scatter plot with explanations

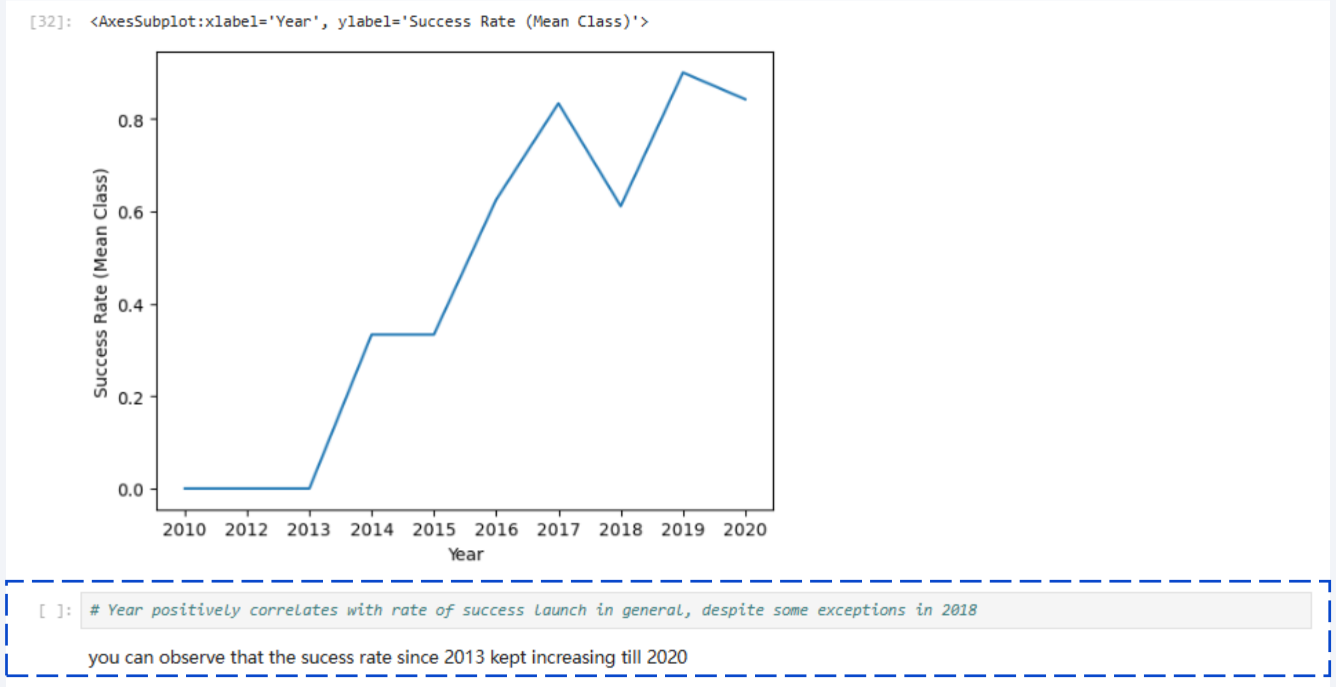


Launch Success Yearly Trend

- Line chart of yearly average success rate



- Screenshot of the scatter plot with explanations



All Launch Site Names

- Find the names of the unique launch sites

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

- Short explanations:

a) As result shows, there are 4 sites with unique name in the connected database of Spacex DataSet
b) As for the database quiry, call:

```
%sql select distinct Launch_Site from SPACEXTABLE
```

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

- Short explanations:

a) As result shows, there should be not less than 5 records where launch sites begin with `CCA` in the connected database of Spacex DataSet

b) As for the database quiry, call:

```
%sql select * from SPACEXTABLE  \
where Launch_Site like 'CCA%' limit 5
```

Date	Time (UTC)	Booster_ Version	Launch _Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_ Outcome	Landing_ Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

Total_Payload_Mass
45596.0

- Short explanations:

a) As result shows, 45596.0 (kg) is the total payload carried by boosters from NASA in the connected database of Spacex DataSet

b) As for the database quiry, call:

```
%sql select round(sum(PAYLOAD_MASS__KG_), 3) as Total_Payload_Mass from SPACEXTABLE  \
where Customer='NASA (CRS)'
```

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

Average_Payload_Mass
2534.667

- Short explanations:

a) As result shows, 2534.667 (kg, rounded to 3 decimal places) is the average payload mass carried by booster version F9 v1.1 in the connected database of Spacex DataSet

b) As for the database quiry, call:

```
%sql select round(avg(PAYLOAD_MASS__KG_), 3) as Average_Payload_Mass from SPACEXTABLE  \
where Booster_Version like 'F9 v1.1%'
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

First_Date
2015-12-22

- Short explanations:

a) As result shows, 2015-12-22 (date) is the first successful landing outcome on ground pad in the connected database of Spacex DataSet, which may set a remarkable day in industry history
b) As for the database quiry, call:

```
%sql select min(Date) as First_Date from SPACEXTABLE where Landing_Outcome='Success (ground pad)'
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

- Short explanations:

a) As result shows, there are 4 unique names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 in the connected database of Spacex DataSet

b) As for the database quiry, call:

```
%sql select distinct Payload from SPACEXTABLE where Landing_Outcome=  \
'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

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

- Short explanations:

a) As result shows, there are 4 typies of mission outcomes with total number summarized in the connected database of Spacex DataSet, among which there are 100 successful missions and 1 failure mission

b) As for the database quiry, call:

```
%sql select Mission_Outcome, count(*) as Total_Number from SPACEXTABLE group by Mission_Outcome
```

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

- Short explanations:

a) As result shows, there are 12 unique names of the booster which have carried the maximum payload mass in the connected database of Spacex DataSet; namely, the payload mass these boosters have carried are equal and top the list

b) As for the database quiry, call:

```
%sql select distinct Booster_Version from SPACEXTABLE where  \
PAYLOAD_MASS__KG_  \
=(select PAYLOAD_MASS__KG_ from SPACEXTABLE order by  \
PAYLOAD_MASS__KG_ desc limit 1)
```

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

Month_Names	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

- Short explanations:

a) As result shows, there are 2 records of failed landing_outcomes in drone ship in year 2015 in the connected database of Spacex DataSet; here lists their booster versions, launch site names and month

b) As for the database quiry, call:

```
%sql select substr(Date,6,2) as Month_Names, Booster_Version, Launch_Site from SPACEXTABLE  \
where Landing_Outcome='Failure (drone ship)' and substr(Date,0,5)='2015'
```

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

- Short explanations:

a) As result shows, there are 8 typies of landing outcomes with total number summarized in the connected database of Spacex DataSet,; here lists the landing outcomes with their total number in descending order

b) As for the database quiry, call:

```
%sql select Landing_Outcome, count(*) as Count from  \
(select Landing_Outcome from SPACEXTABLE where  \
Date between '2010-06-04' and '2017-03-20’)  \
group by Landing_Outcome order by count desc
```

Landing_Outcome	Count
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

Launch Sites in Global Scale

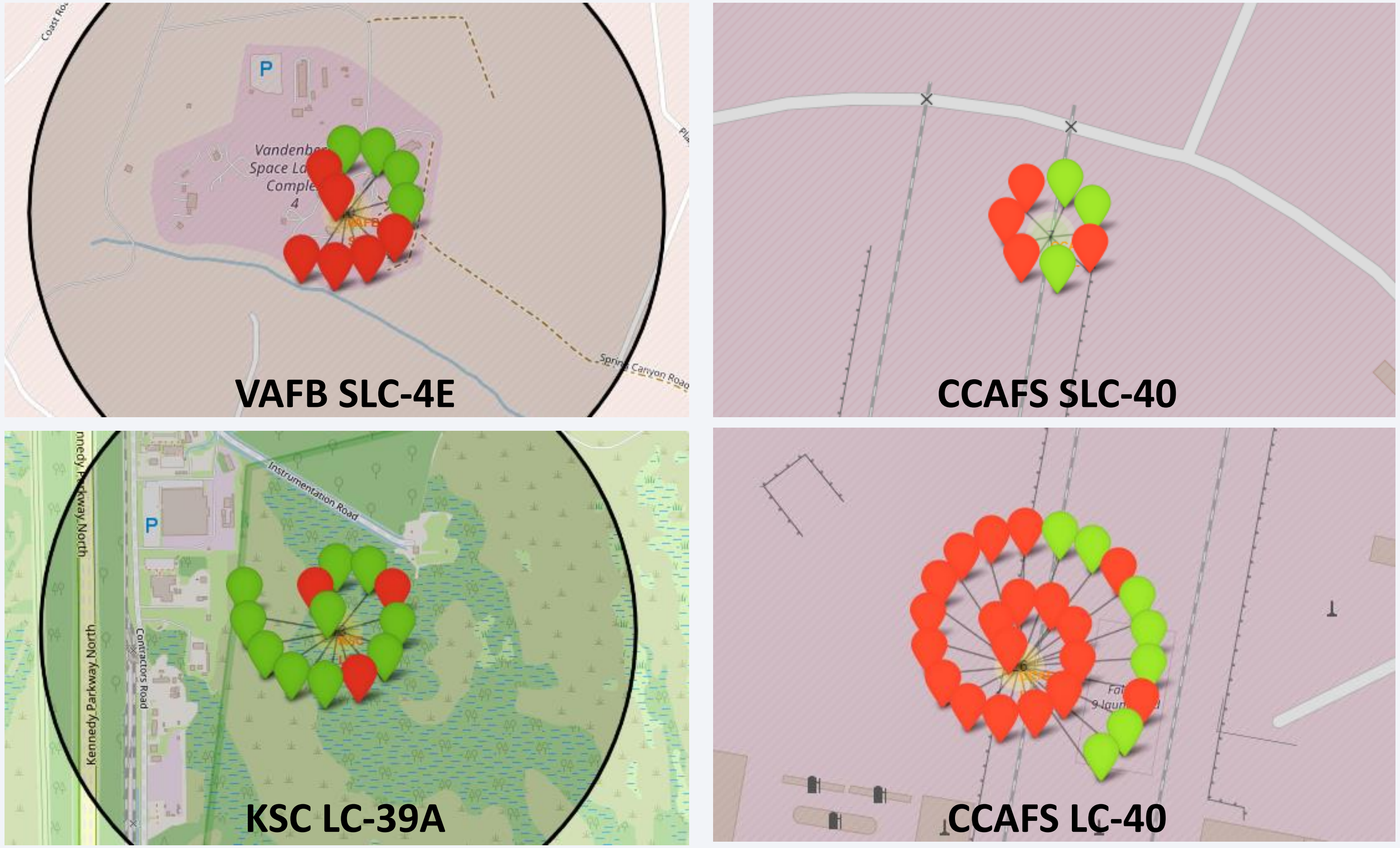
- Generated folium map and screenshot to include all launch sites' location markers on a global map

- Important elements and findings:

a) From global view, each launch site of Spacex is in proximity to the Equator line and the coast
b) Compared with NASA launch site, rules in launch site selection of Spacex is similar to that of NASA
c) Some sites are close to each other, such as CCAFS SLC-40 and CCAFS LC-40

Launch Outcomes of Each Site

Fail launch (Class=0)   Success launch (Class=1)



- Folium map and screenshot to show the color-labeled launch outcomes on the map

- Important elements and findings:

a) More launches in site CCAFS SLC 40 and less in site VAFB SLC 4E
b) Higher launch success ratios in site KSC LC-39A, the rest at similar level

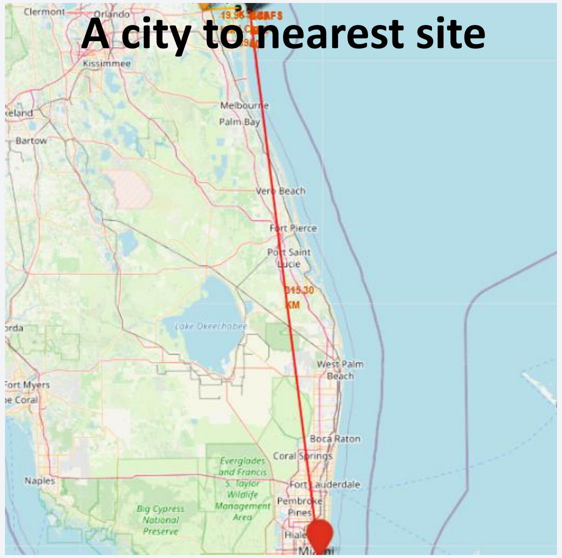
Supporting Elements to Site Selections

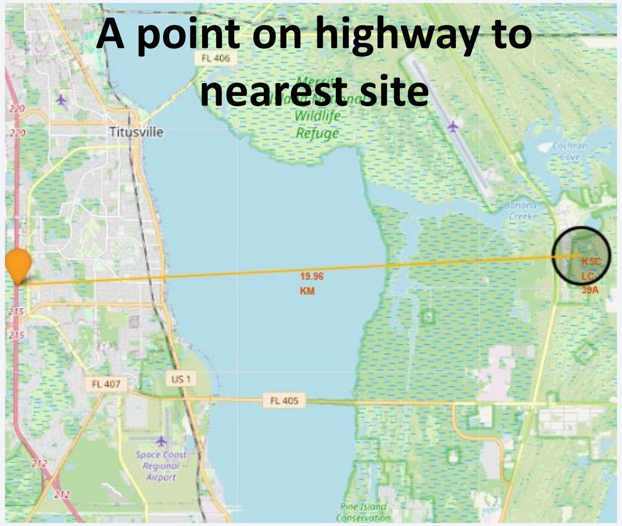
- Generated folium map and screenshot of a selected launch site to its proximities with distance calculated and displayed

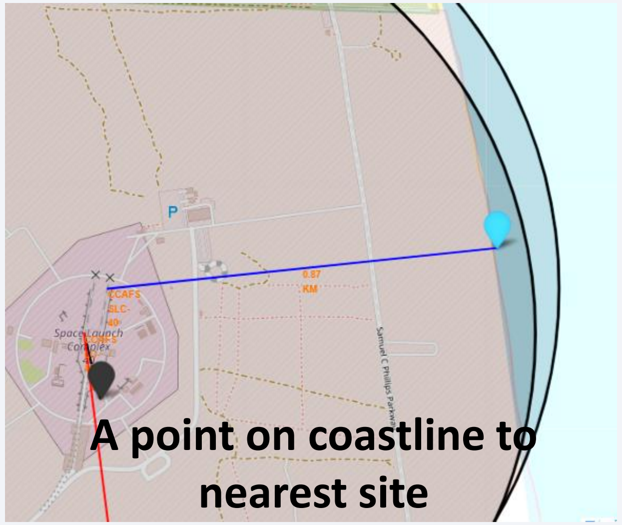
- Important elements and findings:

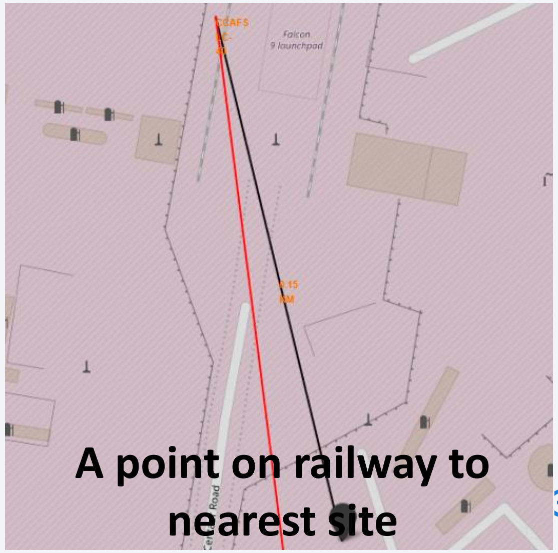
a) Without loss of generality, the distance to launch site follows: railways < coastline < highways < cities

b) As an explanation, concerns on special transportation, dense-population security and operation convenience support decision on site selection of launch









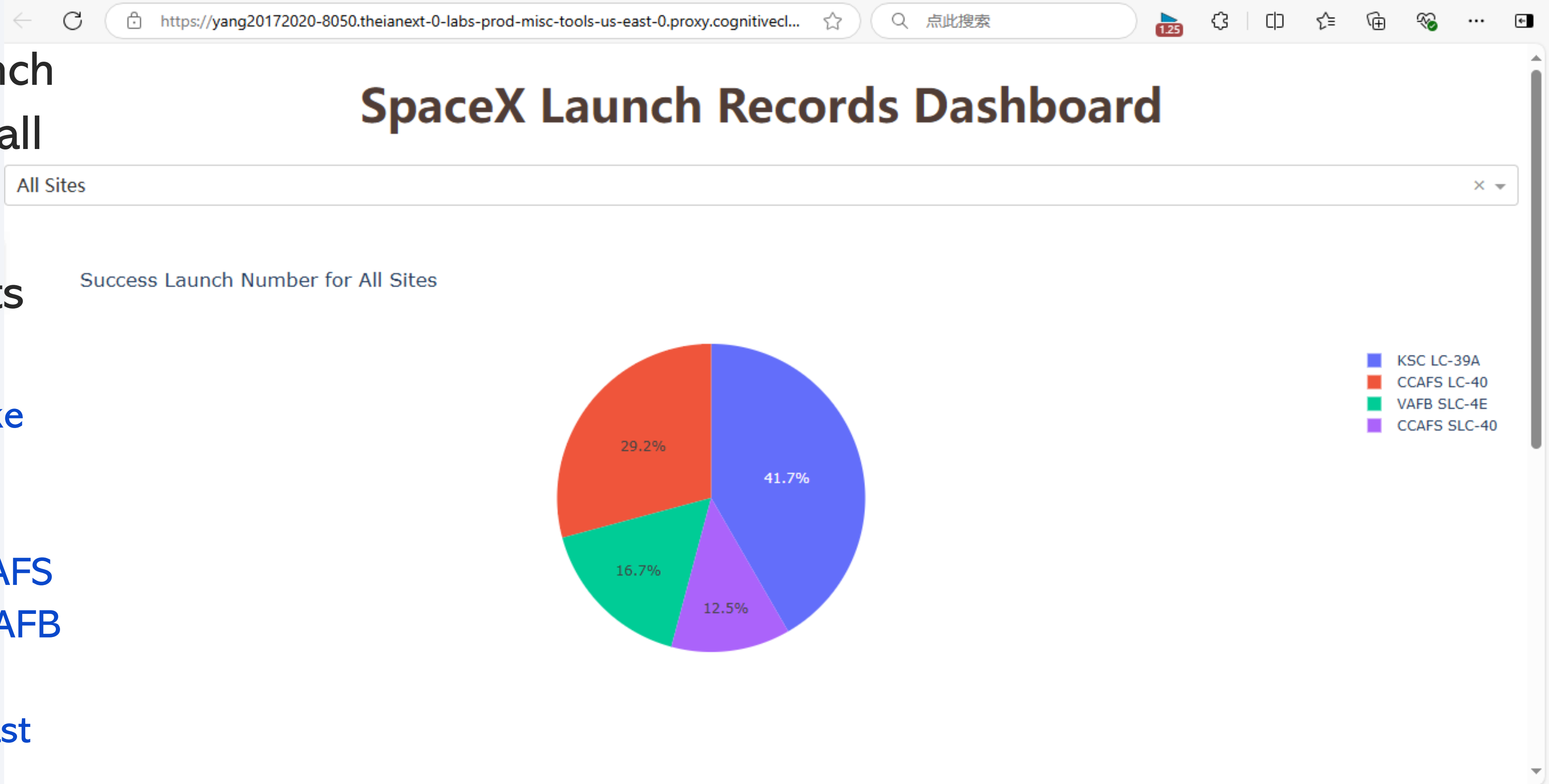
Section 4

Build a Dashboard with Plotly Dash

Launch Success Count for All Sites

- Screenshot of launch success count for all sites, in a piechart
- Important elements and findings:

a) KSC LC-39A has make most launch success among 4 sites in the source csv data, CCAFS LC-40 the second, VAFB SLC-4E the third and CCAFS SLC-40 the last



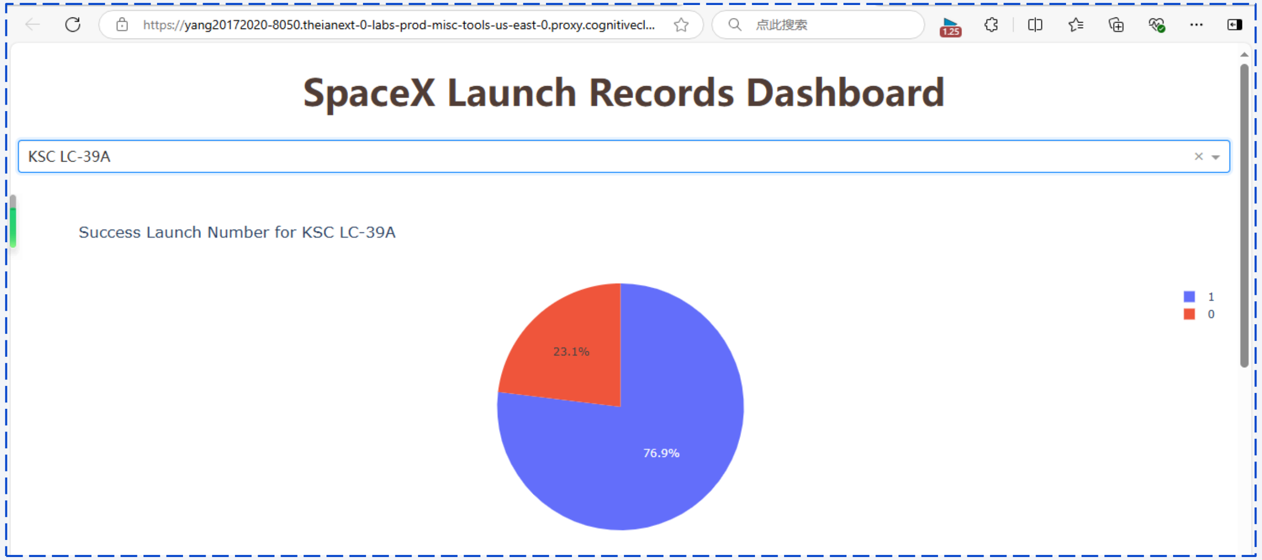
Comparison of Launch Success Ratios

- Important elements and findings:

a) KSC LC-39A is the launch site with highest launch success ratio among 4 sites in the source csv data, CCAFS LC-40 the second

b) KSC LC-39A has done the most successful jobs both in ratio and amount

- Screenshot of the piechart for the launch site with highest launch success ratio



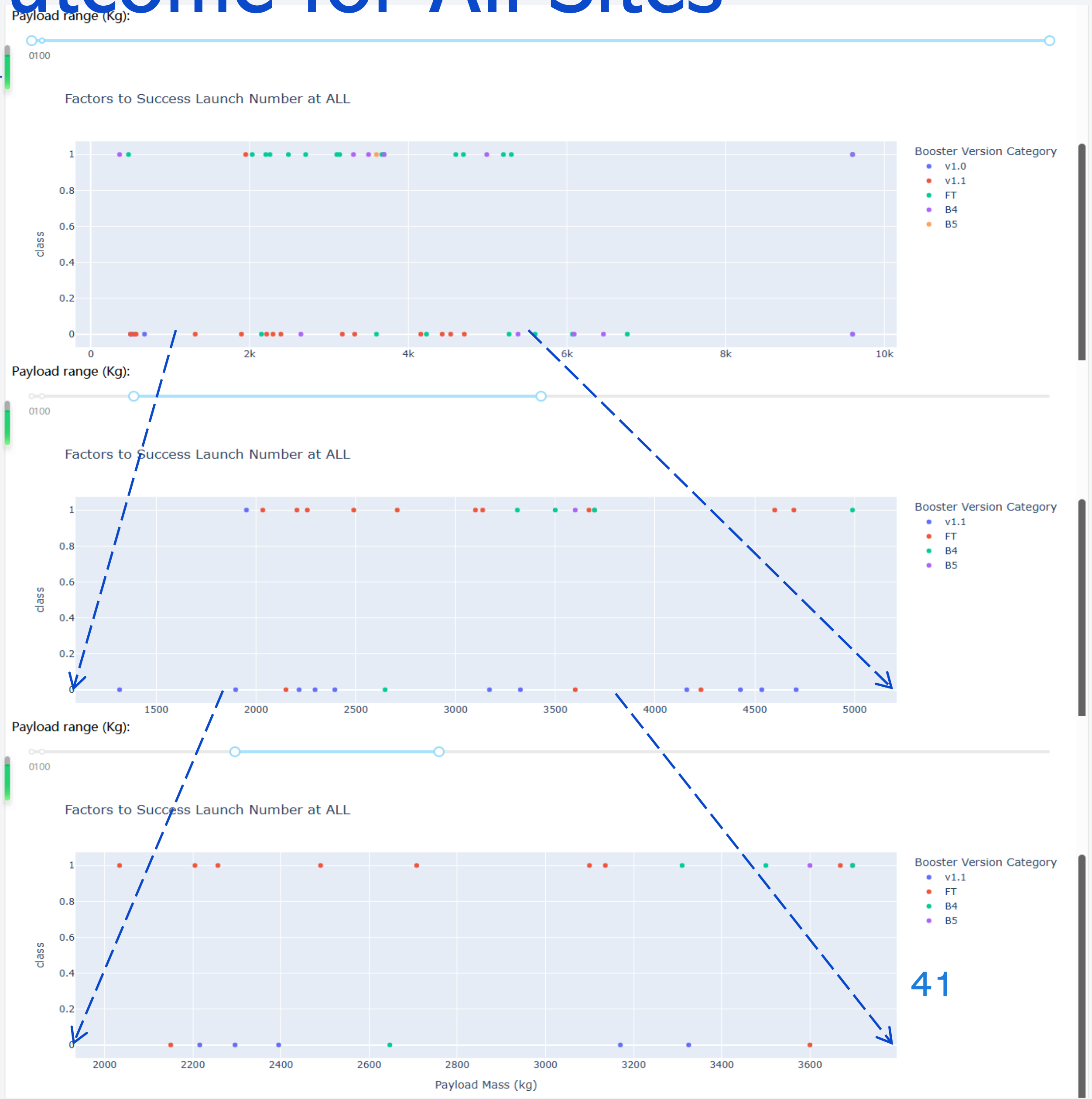
Payload for Best Launch Outcome for All Sites

- Screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- Important elements and findings:

a) In the whole range, payload (kg) with higher launch success ratios appears at about [1100, 5200] for all sites, which is set as subrange1

b) In the subrange1, payload (kg) with higher launch success ratios appears at about [1950, 3800] for all sites, from which it is most likely to pick out the payload for best launch outcome



Section 5

Predictive Analysis (Classification)

Variables to Perform Classification

- The data collected and wrangled (dropped if irrelative, set dummy where necessary, casted to float64 and standardized) has potentials to predict launch outcomes

- We set the dependent variable as:

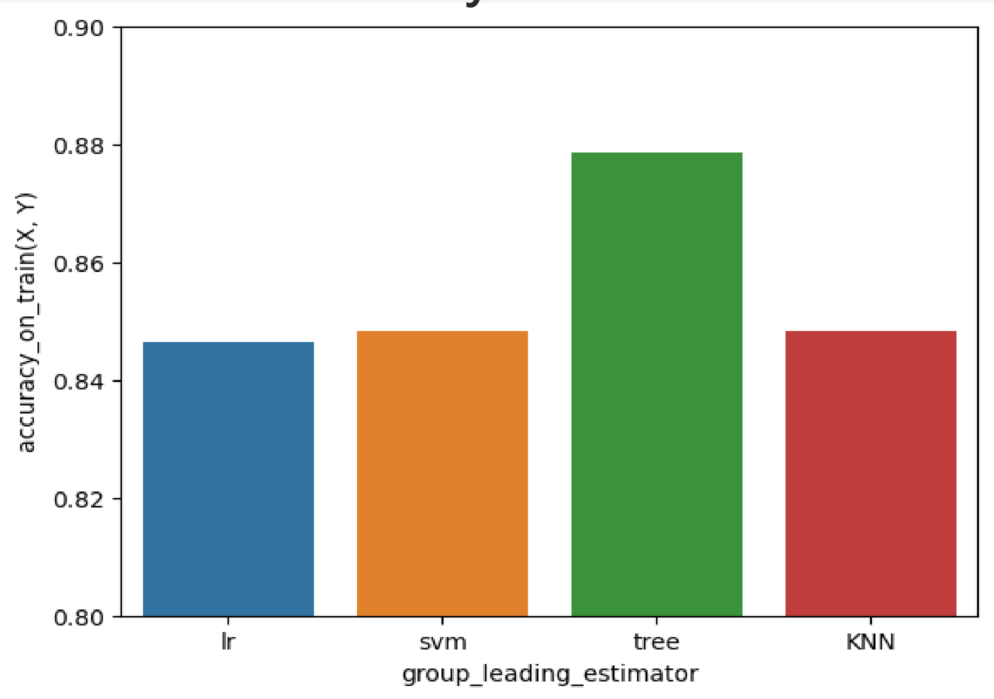
Variable Name	Original Type
Outcome	object

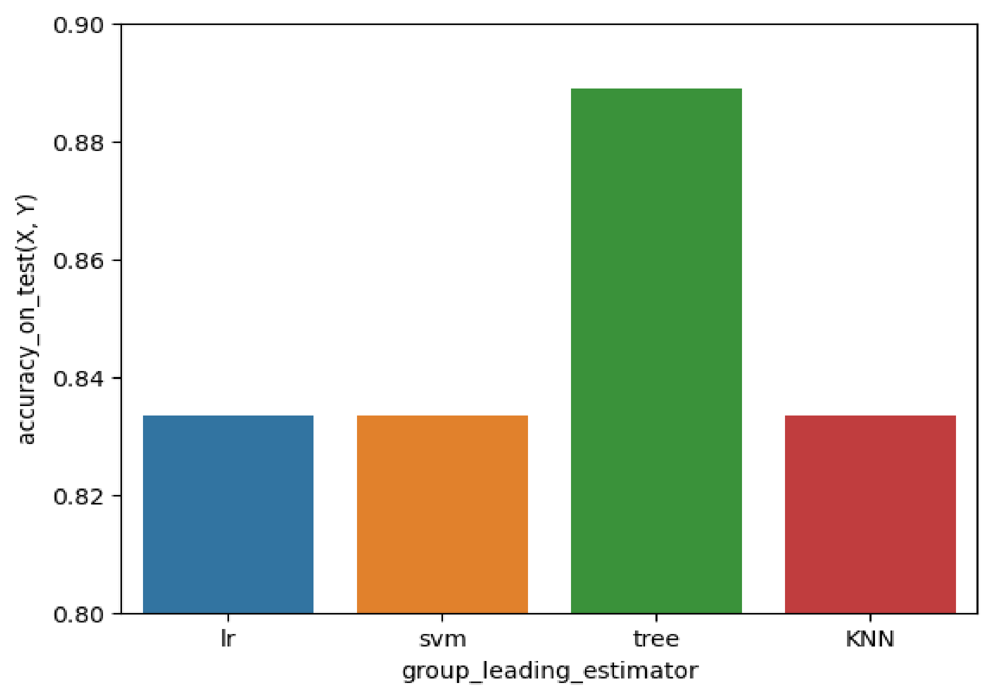
- We set the independent variable as:

Variable Name	Original Type
FlightNumber	int64
PayloadMass	float64
Flights Block	float64
ReusedCount	int64
Orbit	object
LaunchSite	object
LandingPad	object
Serial	object
GridFins	bool
Reused	bool
Legs	bool

Classification Accuracy

- Built model accuracy for all built classification models in a bar chart





- Model ‘tree’ has the highest classification accuracy so far among the 4 group leaders:

  ‘lr’: logistic regression on {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs‘}

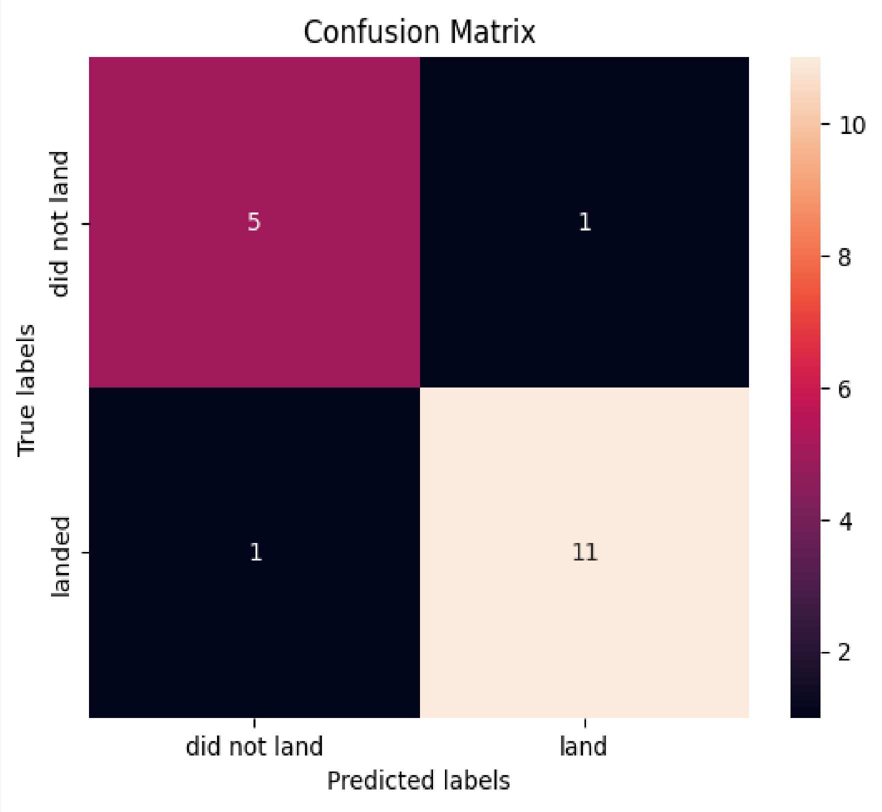
  ‘svm’: support vector machine on {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid’}

  ‘tree’: decision tree classifier on {'criterion': 'entropy', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random’}

  ‘KNN’: k nearest neighbors on {'algorithm': 'auto', 'n_neighbors': 10, 'p’: 1}

Confusion Matrix

- Confusion matrix of the best performing model 'tree'



- Explanations:

a) Each confusion matrix row shows the actual true labels in test set, and the columns show the predicted labels by classifier.

b) Interpreting: true positives=11, false positive=1, true negative=5, false negative=1; For 'did not land' cases, Precision=5/(5+1)=0.833, Recall=5/(5+1)=0.833, F1-score=0.833; For 'land' cases, Precision=11/(11+1)=0.917, Recall=11/(11+1) =0.917, F1-score=0.917; So Avg Accuracy=(0.833+0.917)/2=0.875, close to 1 in high level

Conclusions

- EDA does tell some narrative stories: Flight Num positively correlates with launch success ratios at site CCAFS SLC 40, Orbit generally correlates with launch success ratios to a certain extent, Year positively generally correlates with launch success ratios
- Site selection of launch may depend on such major supporting elements as latitude (proximity to equator), proximity to railways and coastline (and probably highways), buffering distance to cities, etc., which may shed lights on infrastructure works
- Such variables as LaunchSite and PayloadMass produce unignorable influence on launch outcomes, despite the different scales and routes they may perform, which may provide clues to streamline technological patterns
- Launch outcome can be predicted based on FlightNumber, PayloadMass, Flights Block, ReusedCount, Orbit (dummy), LaunchSite (dummy), LandingPad (dummy), Serial (dummy), GridFins (binary), Reused (binary), Legs (binary) in high accuracy using model 'tree', which may set foundation for quantitative decision process

Appendix

- Notebook outputs:

a) https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb
b) https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb
c) https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data wrangling.ipynb
d) https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/edadataviz.ipynb
e) https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb
f) https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb
g) https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine Learning Prediction_Part_5.ipynb

- Python code snippets:

a) https://github.com/yang20172020/Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py

Thank you!