IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Houman Kazemzadeh
September, 2nd, 2024

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - By using requests library, the Json file was obtained from the database, and was converted to a data frame using pandas. Only Falcon 9 data was maintained for future predictions and wrangling.
- Perform data wrangling
  - By Importing NumPy and pandas, value_counts, isnull, and fillna was performed.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Sklearn (preprocessing, train_test_split, GridSearchCV, Logistic Regression, Support Vector Machine, Decision Tree, KNN

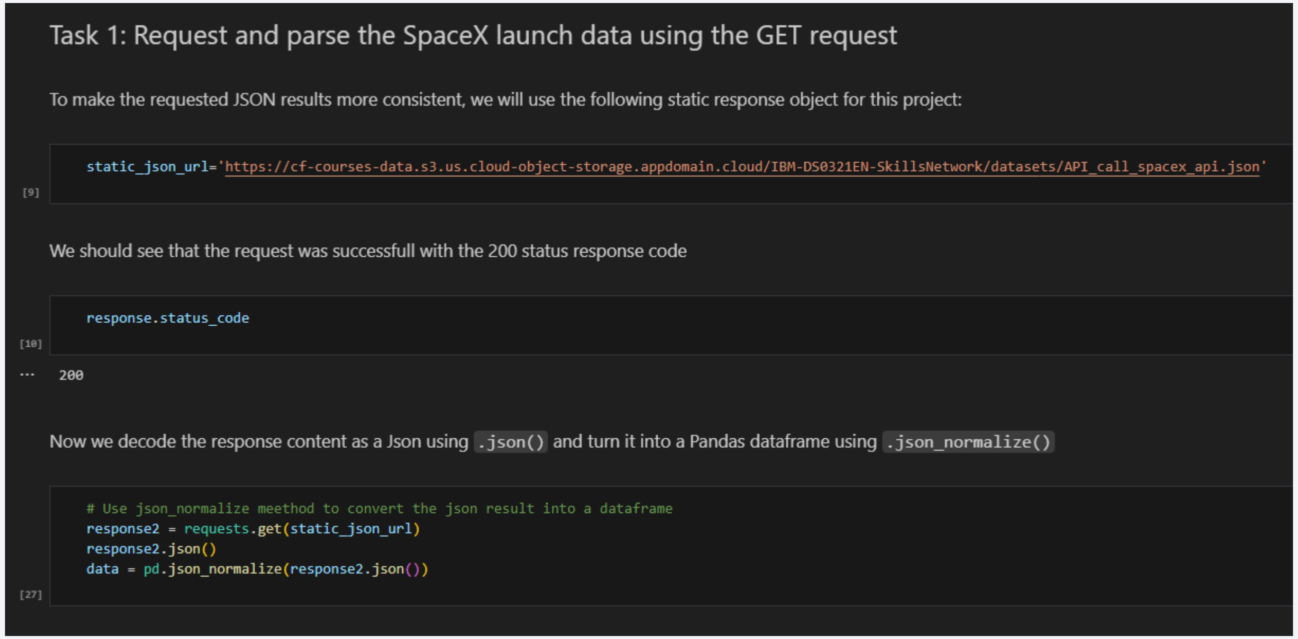
Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts
- By using requests library, the Json file was obtained from the database, and was converted to a data frame using pandas. Only Falcon 9 data was maintained for future predictions and wrangling.

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose



Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[9]
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
[10]
response.status_code
```

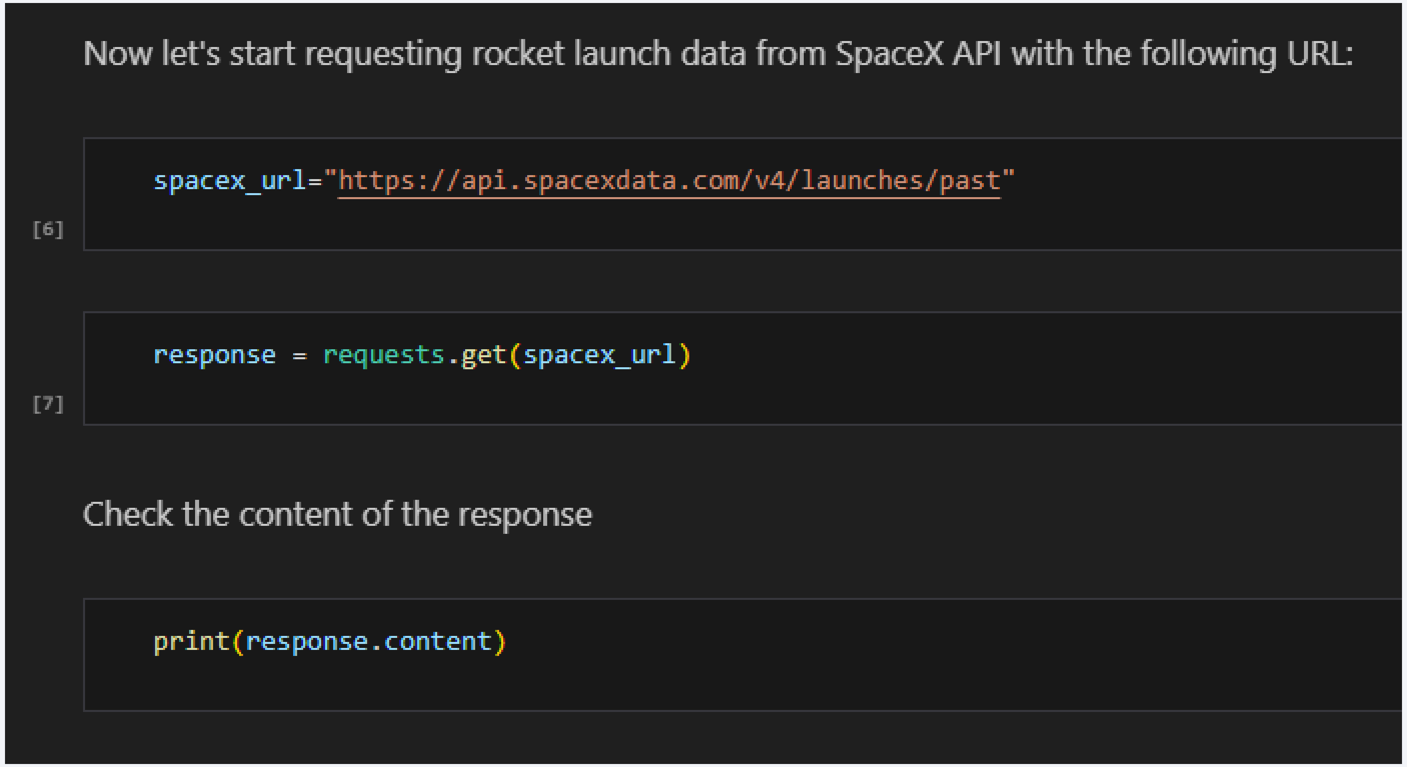
```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[27]
# Use json_normalize meethod to convert the json result into a dataframe
response2 = requests.get(static_json_url)
response2.json()
data = pd.json_normalize(response2.json())
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



Now let's start requesting rocket launch data from SpaceX API with the following URL:

[6]

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

[7]

```
response = requests.get(spacex_url)
```

Check the content of the response

```
print(response.content)
```

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose
- By Importing NumPy and pandas, value_counts, isnull, and fillna was performed.

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- Pandas, NumPy and Seaborn were imported. Scatterplot, Bar plot, and line plot were illustrated to better understand the interactions between features and the results.

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
- Did not complete it

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
- Did not Complete it

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose
- Did not complete it

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
- Sklearn (preprocessing, train_test_split, GridSearchCV, Logistic Regression, Support Vector Machine, Decision Tree, KNN

Results

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

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

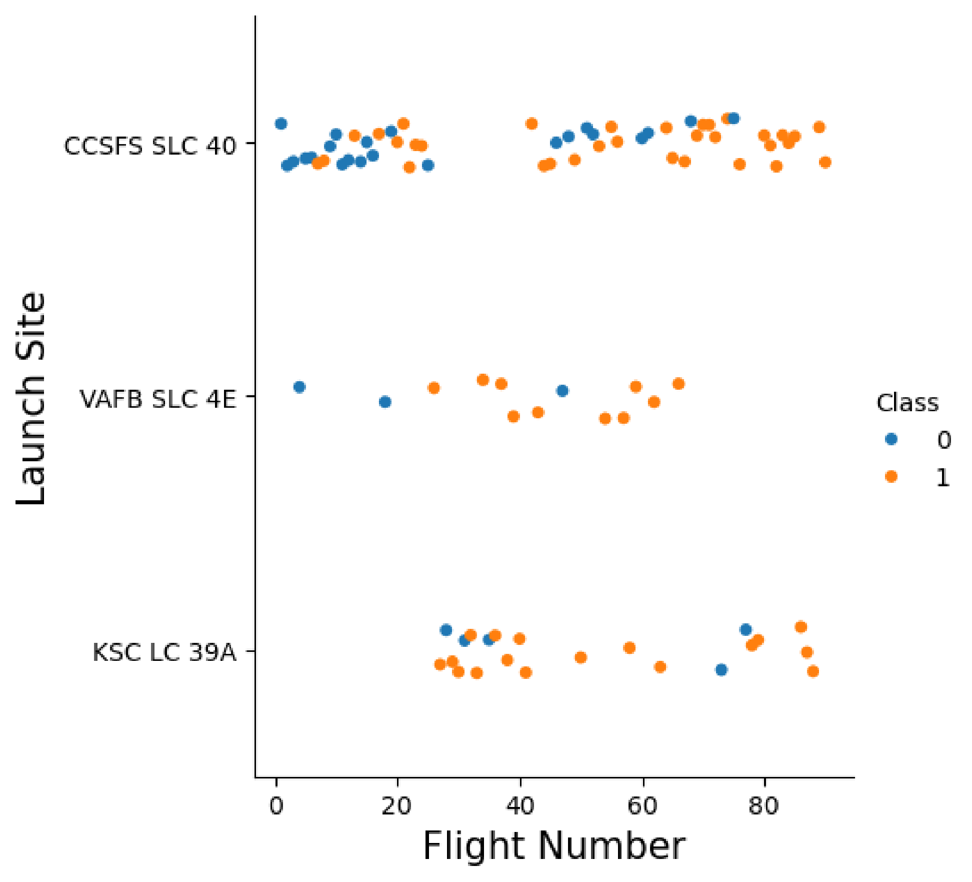
- Show a scatter plot of Flight Number vs. Launch Site

- Show the screenshot of the scatter plot with explanations

- Launch Site Distribution: The majority of launches seem to have occurred at CCSFS SLC 40.VAFB SLC 4E has a smaller number of launches. KSC LC 39A has the fewest launches.

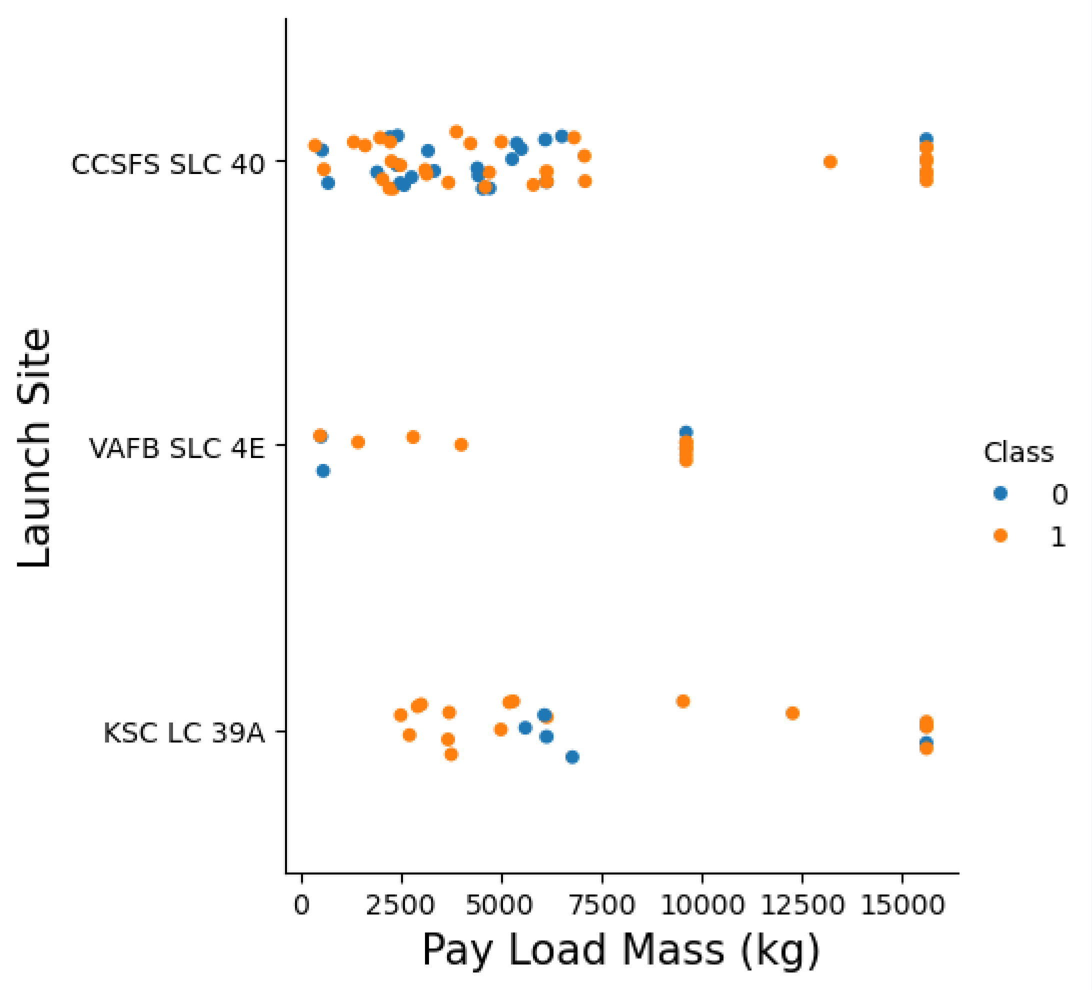
- Class Distribution: Both blue and orange dots are present at all three launch sites, suggesting that both classes of launches have occurred at each location. The relative proportion of blue and orange dots at each site might indicate different success rates or other factors associated with the "Class" variable.

- Trend with Flight Number: There doesn't appear to be a clear trend between Flight Number and Launch Site. The dots are scattered across the x-axis for each launch site.



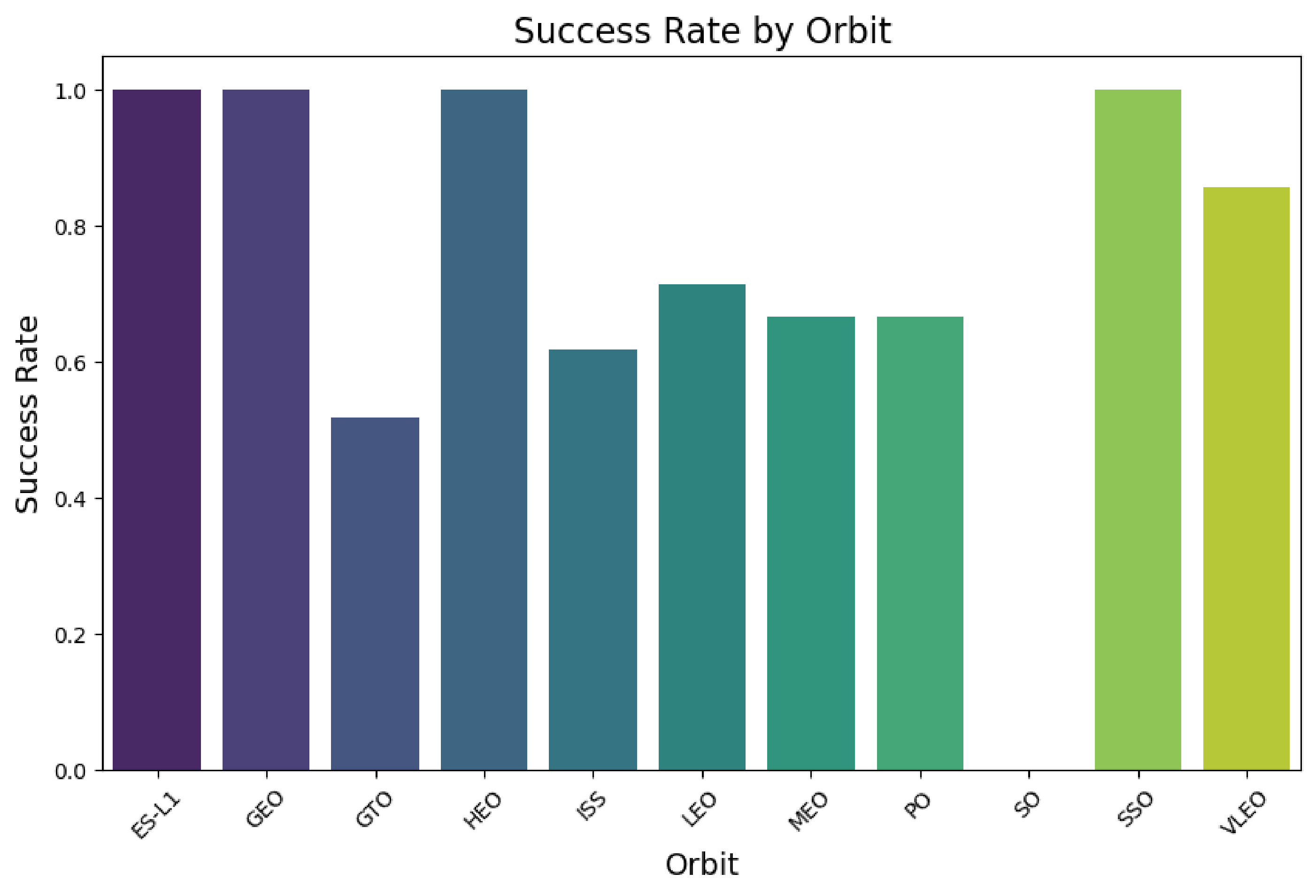
Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- Show the screenshot of the scatter plot with explanations
- Payload Mass Distribution:The majority of launches have payloads below 10,000 kg.There are a few launches with significantly higher payload masses, particularly at CCSFS SLC 40.The distribution of payload masses varies across the launch sites.
- Launch Site Distribution:CCSFS SLC 40 has the highest number of launches, with a wide range of payload masses.VAFB SLC 4E has a smaller number of launches, mostly concentrated in the lower payload mass range.KSC LC 39A has the fewest launches, with a mix of payload masses.
- Class Distribution:Both blue and orange dots are present at all three launch sites, suggesting that both classes of launches have occurred at each location.The relative proportion of blue and orange dots at each site might indicate different success rates or other factors associated with the "Class" variable.
- Relationship between Payload Mass and Launch Site:There seems to be a slight correlation between payload mass and launch site. For example, CCSFS SLC 40 has launches with a wider range of payload masses compared to VAFB SLC 4E.



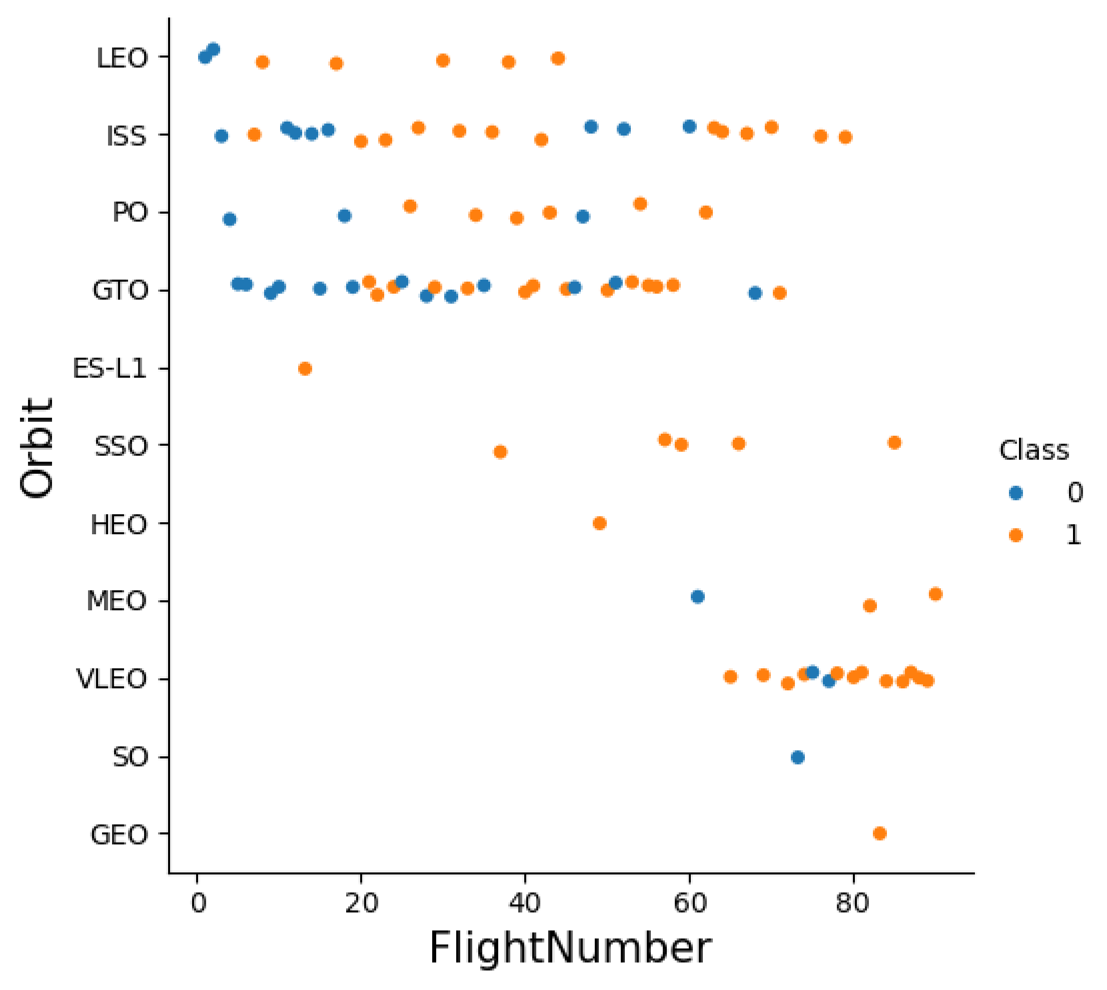
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations
- Highest Success Rates:ES-L1, GEO, and HEO have the highest success rates, approaching 1.0.SSO also has a relatively high success rate.
- Lowest Success Rates:ISS and VLEO have the lowest success rates.
- Variability:There is a noticeable variation in success rates among the different orbits. Some orbits have consistently high success rates, while others have lower or more inconsistent performance.



Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations
- Orbit Distribution:LEO, ISS, and PO have the highest number of launches.Other orbits, such as VLEO, SO, and GEO, have fewer launches.
- Class Distribution:Both blue and orange dots are present for most orbits, suggesting that both classes of launches have occurred at different altitudes.The relative proportion of blue and orange dots for each orbit might indicate different success rates or other factors associated with the "Class" variable.
- Trend with Flight Number:There doesn't appear to be a clear trend between Flight Number and Orbit. The dots are scattered across the x-axis for each orbit.



Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

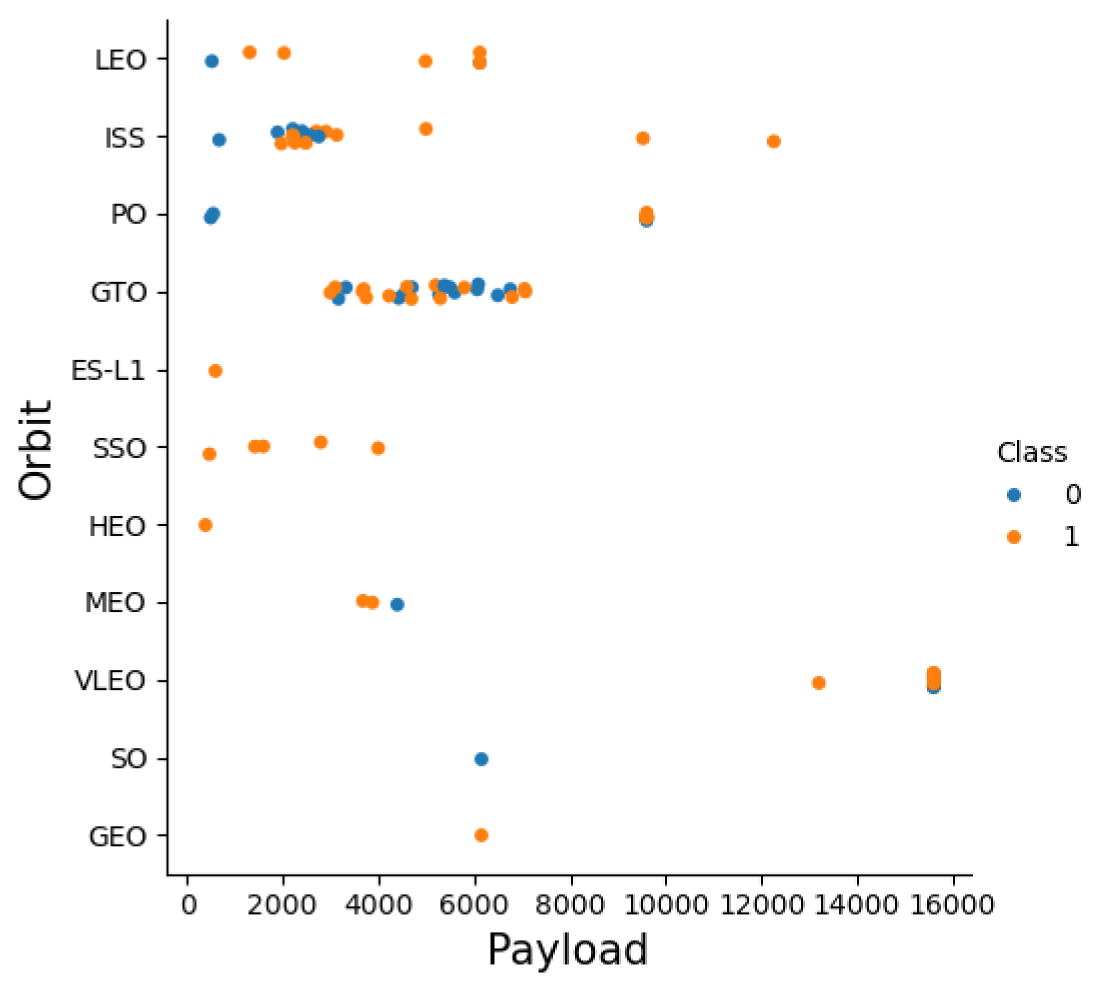
- Show the screenshot of the scatter plot with explanations

- Payload Distribution:The majority of launches have payloads below 10,000 kg.There are a few launches with significantly higher payloads, particularly in the LEO and GTO orbits.The distribution of payload masses varies across the different orbits.

- Orbit Distribution:LEO, ISS, and PO have the highest number of launches, with a wide range of payload masses.Other orbits, such as VLEO, SO, and GEO, have fewer launches, often with more limited payload capacities.

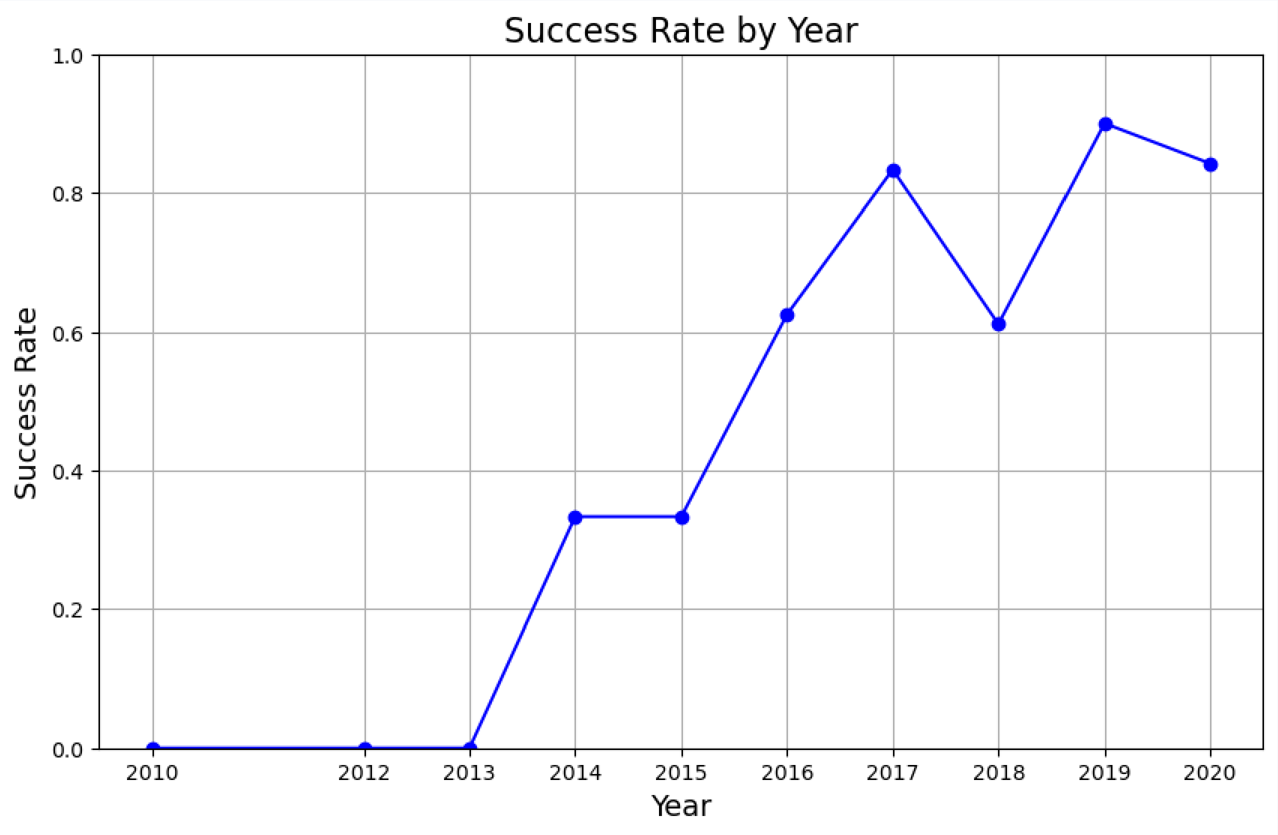
- Class Distribution:Both blue and orange dots are present for most orbits, suggesting that both classes of launches have occurred at different altitudes.The relative proportion of blue and orange dots for each orbit might indicate different success rates or other factors associated with the "Class" variable.

- Relationship between Payload and Orbit:There seems to be a correlation between payload and orbit. For example, LEO and GTO orbits often have higher payload capacities compared to other orbits.



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations
- Overall Trend:There is a clear upward trend in success rate over the years.The success rate starts low in 2010 and gradually increases, reaching its peak in 2019.In 2020, there is a slight decrease in success rate.
- Specific Years:The years 2010, 2011, 2012, and 2013 had very low success rates.A significant improvement occurred between 2013 and 2014.There were fluctuations in success rate between 2015 and 2019.



All Launch Site Names

- Find the names of the unique launch sites

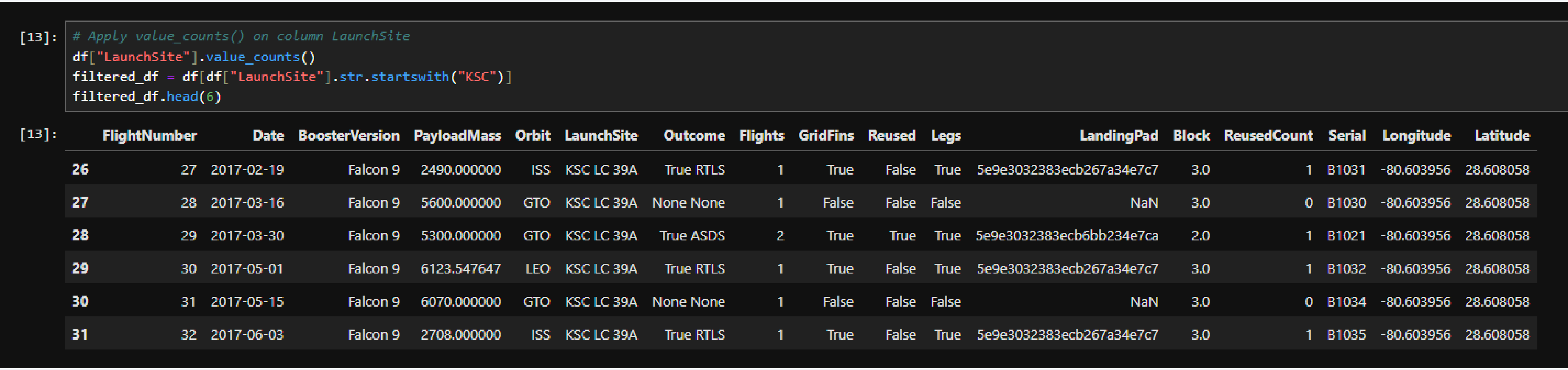
```
LaunchSite
CCSFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: count, dtype: int64
```

- Present your query result with a short explanation here

df["LaunchSite"].value_counts() to see unique values and their counts

Launch Site Names Begin with 'KSC'

- Find 5 records where launch sites' names start with `KSC`
- Present your query result with a short explanation here



[13]:
```
# Apply value_counts() on column LaunchSite
df["LaunchSite"].value_counts()
filtered_df = df[df["LaunchSite"].str.startswith("KSC")]
filtered_df.head(6)
```

[13]:

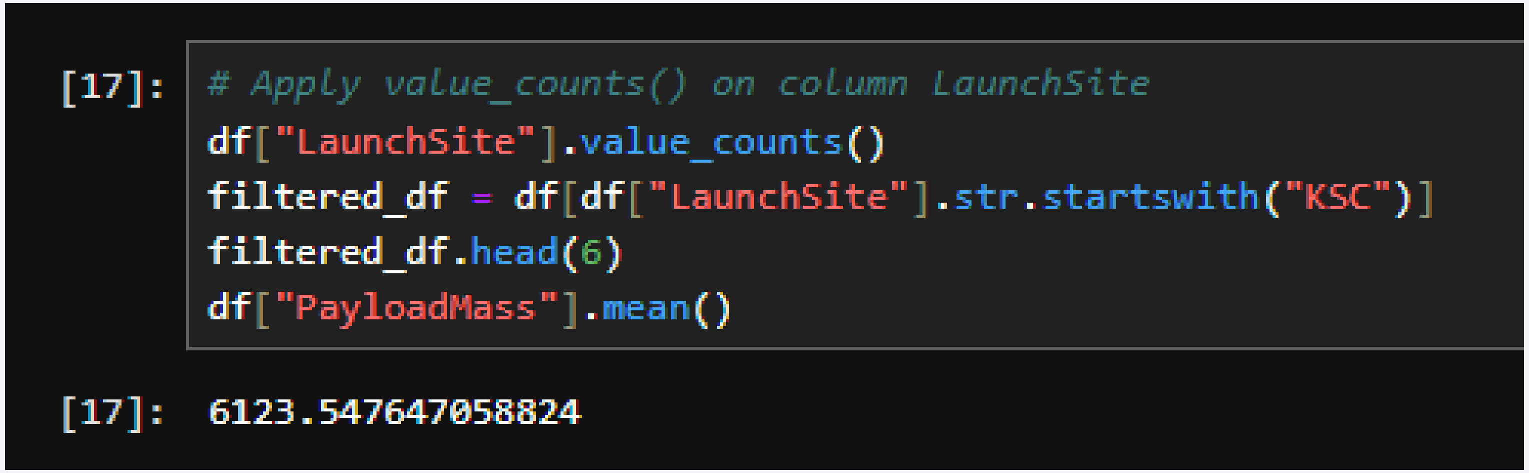
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
26	27	2017-02-19	Falcon 9	2490.000000	ISS	KSC LC 39A	True RTLS	1	True	False	True	5e9e3032383ecb267a34e7c7	3.0	1	B1031	-80.603956	28.608058
27	28	2017-03-16	Falcon 9	5600.000000	GTO	KSC LC 39A	None None	1	False	False	False	NaN	3.0	0	B1030	-80.603956	28.608058
28	29	2017-03-30	Falcon 9	5300.000000	GTO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	2.0	1	B1021	-80.603956	28.608058
29	30	2017-05-01	Falcon 9	6123.547647	LEO	KSC LC 39A	True RTLS	1	True	False	True	5e9e3032383ecb267a34e7c7	3.0	1	B1032	-80.603956	28.608058
30	31	2017-05-15	Falcon 9	6070.000000	GTO	KSC LC 39A	None None	1	False	False	False	NaN	3.0	0	B1034	-80.603956	28.608058
31	32	2017-06-03	Falcon 9	2708.000000	ISS	KSC LC 39A	True RTLS	1	True	False	True	5e9e3032383ecb267a34e7c7	3.0	1	B1035	-80.603956	28.608058

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

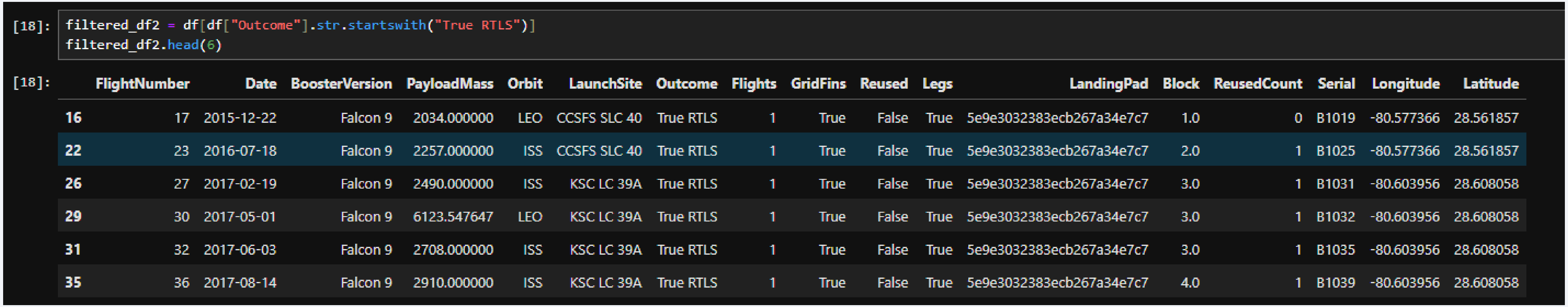


```
[17]: # Apply value_counts() on column LaunchSite
      df["LaunchSite"].value_counts()
      filtered_df = df[df["LaunchSite"].str.startswith("KSC")]
      filtered_df.head(6)
      df["PayloadMass"].mean()

[17]: 6123.547647058824
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on drone ship.Present your query result with a short explanation here



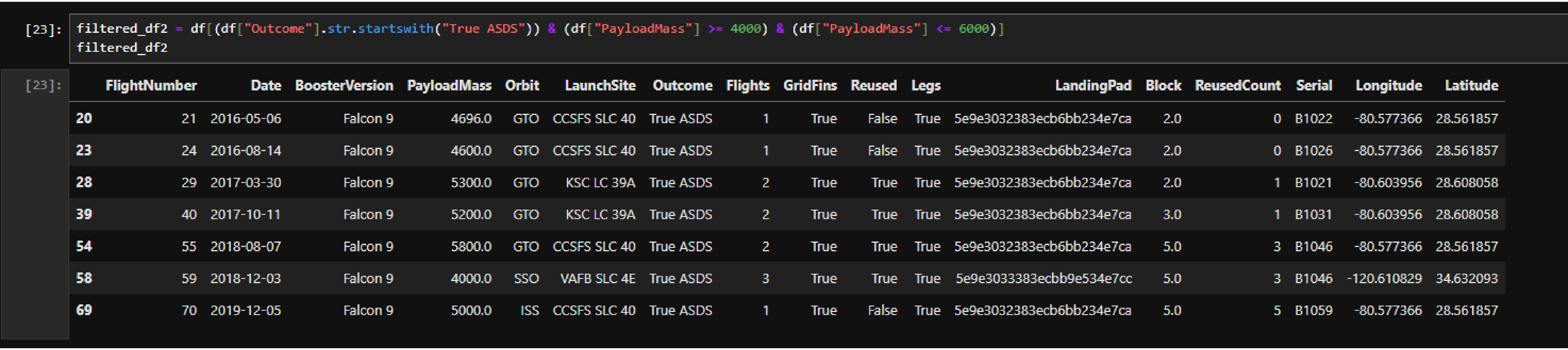
```
[18]: filtered_df2 = df[df["Outcome"].str.startswith("True RTLS")]
      filtered_df2.head(6)
```

[18]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
16	17	2015-12-22	Falcon 9	2034.000000	LEO	CCSFS SLC 40	True RTLS	1	True	False	True	5e9e3032383ecb267a34e7c7	1.0	0	B1019	-80.577366	28.561857
22	23	2016-07-18	Falcon 9	2257.000000	ISS	CCSFS SLC 40	True RTLS	1	True	False	True	5e9e3032383ecb267a34e7c7	2.0	1	B1025	-80.577366	28.561857
26	27	2017-02-19	Falcon 9	2490.000000	ISS	KSC LC 39A	True RTLS	1	True	False	True	5e9e3032383ecb267a34e7c7	3.0	1	B1031	-80.603956	28.608058
29	30	2017-05-01	Falcon 9	6123.547647	LEO	KSC LC 39A	True RTLS	1	True	False	True	5e9e3032383ecb267a34e7c7	3.0	1	B1032	-80.603956	28.608058
31	32	2017-06-03	Falcon 9	2708.000000	ISS	KSC LC 39A	True RTLS	1	True	False	True	5e9e3032383ecb267a34e7c7	3.0	1	B1035	-80.603956	28.608058
35	36	2017-08-14	Falcon 9	2910.000000	ISS	KSC LC 39A	True RTLS	1	True	False	True	5e9e3032383ecb267a34e7c7	4.0	1	B1039	-80.603956	28.608058

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
- Present your query result with a short explanation here



[23]:

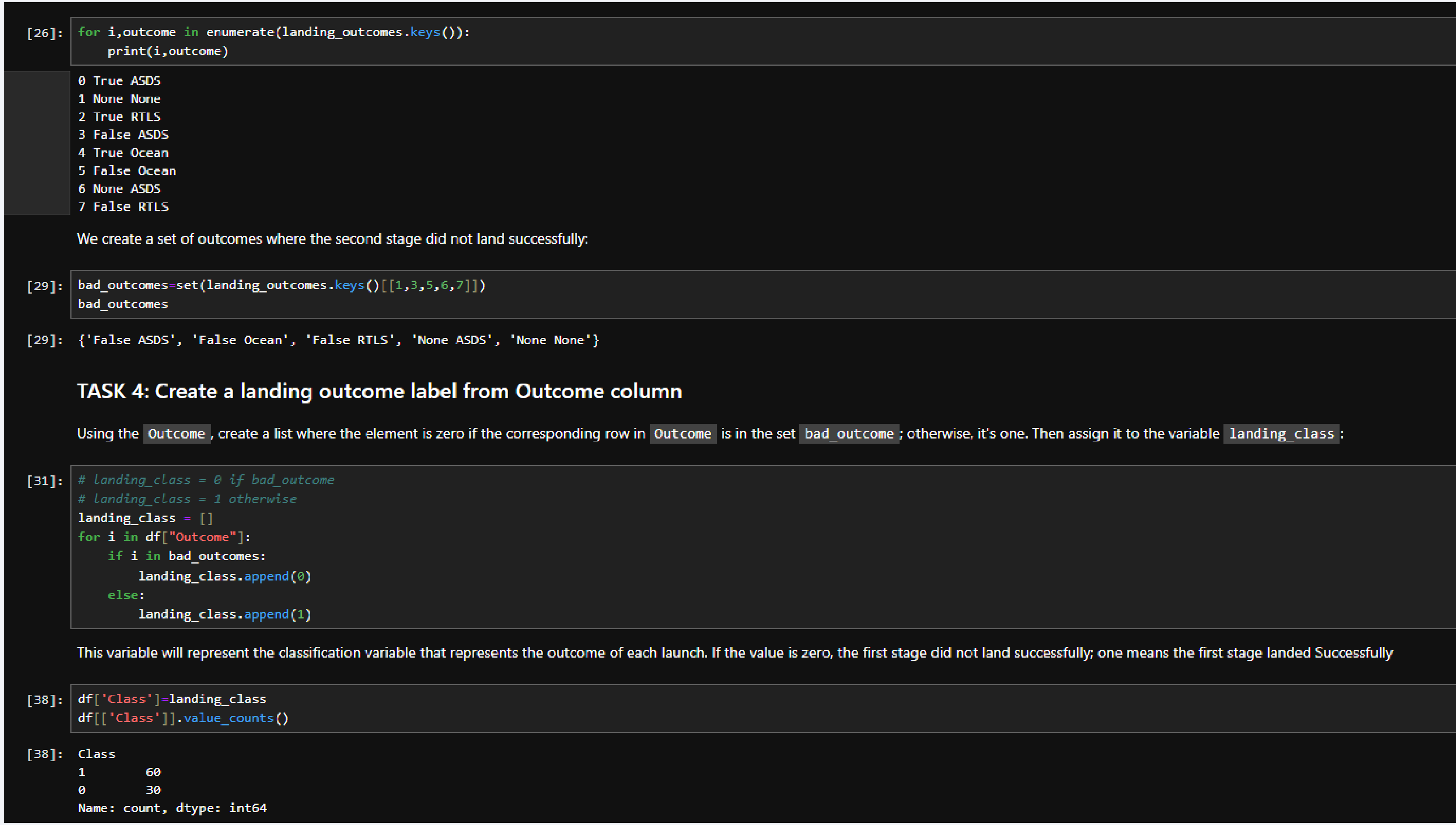
```
filtered_df2 = df[(df["Outcome"].str.startswith("True ASDS")) & (df["PayloadMass"] >= 4000) & (df["PayloadMass"] <= 6000)]
filtered_df2
```

[23]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
20	21	2016-05-06	Falcon 9	4696.0	GTO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	2.0	0	B1022	-80.577366	28.561857
23	24	2016-08-14	Falcon 9	4600.0	GTO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	2.0	0	B1026	-80.577366	28.561857
28	29	2017-03-30	Falcon 9	5300.0	GTO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	2.0	1	B1021	-80.603956	28.608058
39	40	2017-10-11	Falcon 9	5200.0	GTO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	3.0	1	B1031	-80.603956	28.608058
54	55	2018-08-07	Falcon 9	5800.0	GTO	CCSFS SLC 40	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	3	B1046	-80.577366	28.561857
58	59	2018-12-03	Falcon 9	4000.0	SSO	VAFB SLC 4E	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	3	B1046	-120.610829	34.632093
69	70	2019-12-05	Falcon 9	5000.0	ISS	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	5	B1059	-80.577366	28.561857

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here



```
[26]: for i,outcome in enumerate(landing_outcomes.keys()):
          print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

We create a set of outcomes where the second stage did not land successfully:

```
[29]: bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
      bad_outcomes
```

```
[29]: {'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

```
[31]: # landing_class = 0 if bad_outcome
      # landing_class = 1 otherwise
      landing_class = []
      for i in df["Outcome"]:
          if i in bad_outcomes:
              landing_class.append(0)
          else:
              landing_class.append(1)
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

```
[38]: df['Class']=landing_class
      df[['Class']].value_counts()
```

```
[38]: Class
      1        60
      0        30
      Name: count, dtype: int64
```

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

2015 Launch Records

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

- Present your query result with a short explanation here

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

- Present your query result with a short explanation here

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Replace <Folium map screenshot 1> title with an appropriate title

- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map

- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

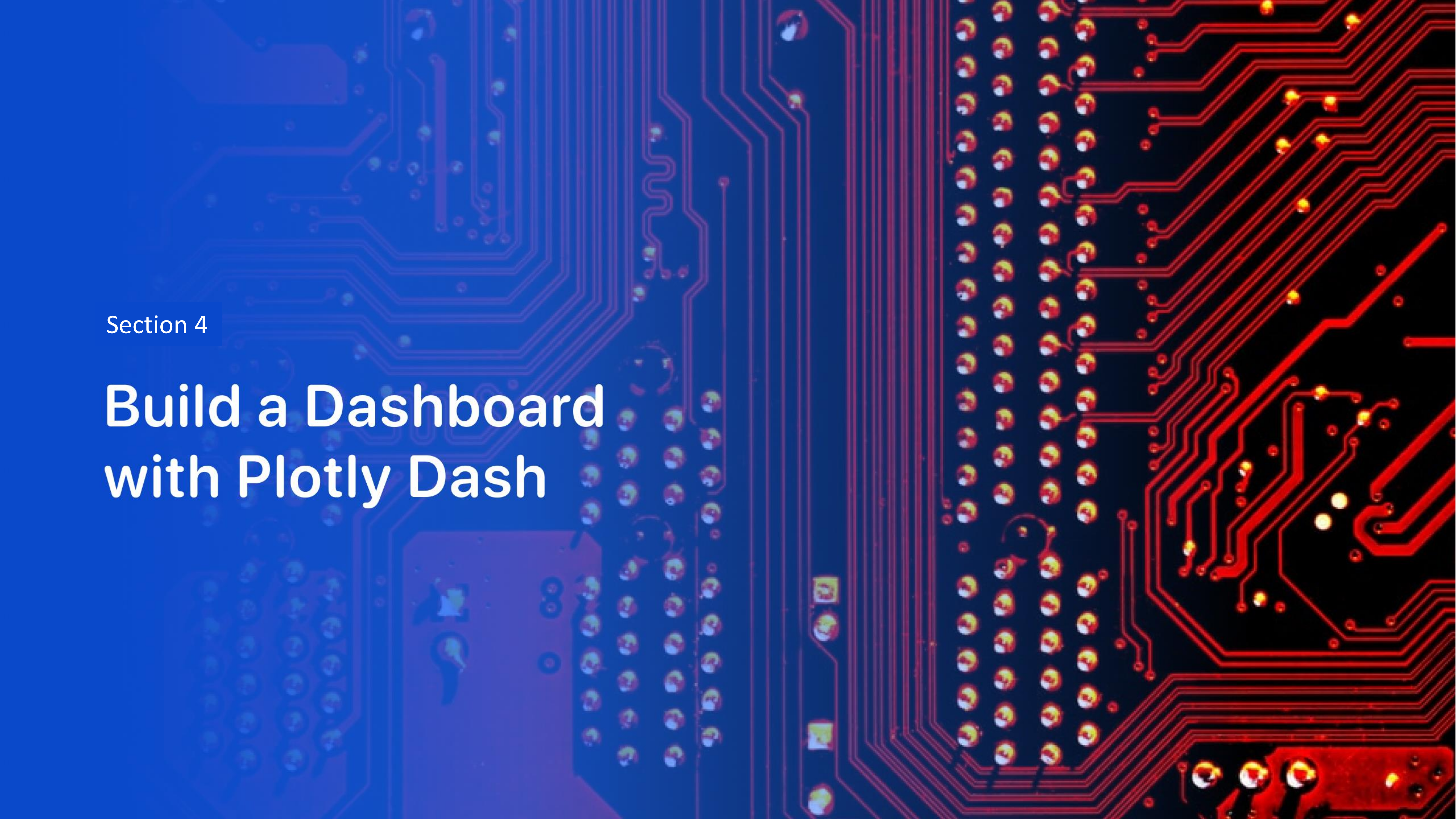
- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot



Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title

- Show the screenshot of the piechart for the launch site with highest launch success ratio

- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title

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

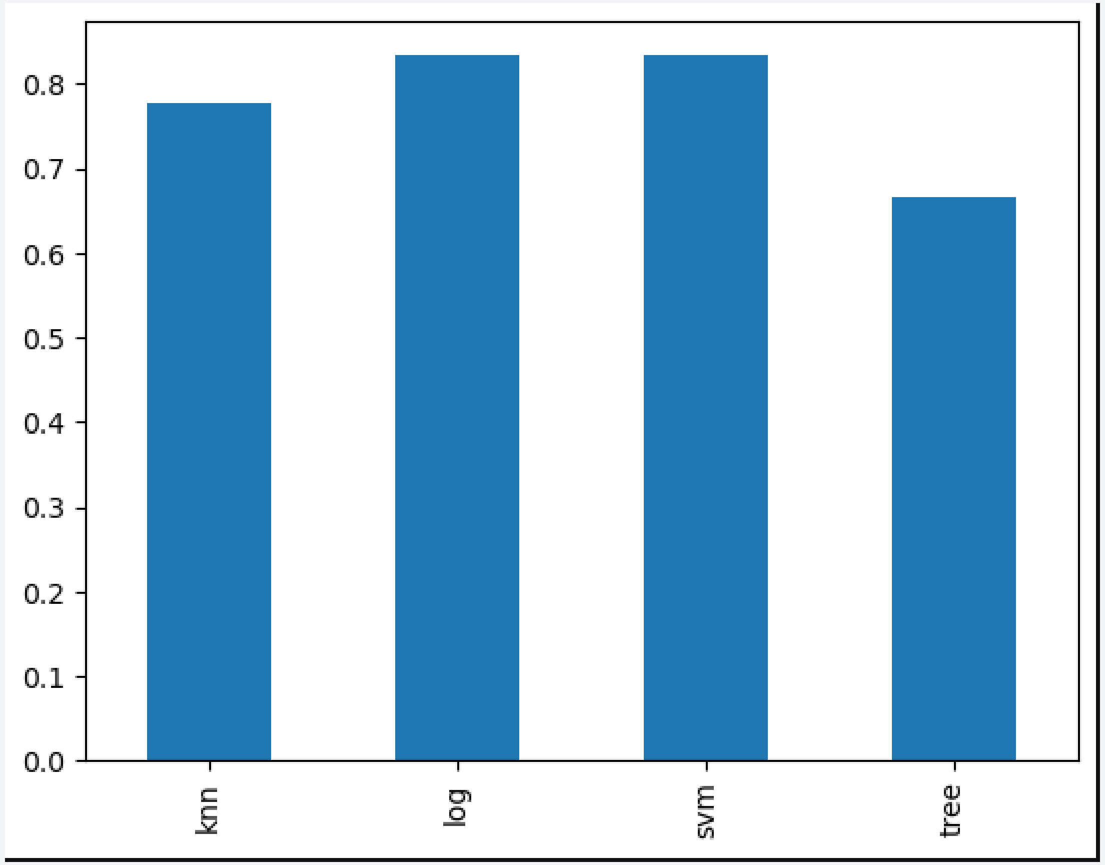
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

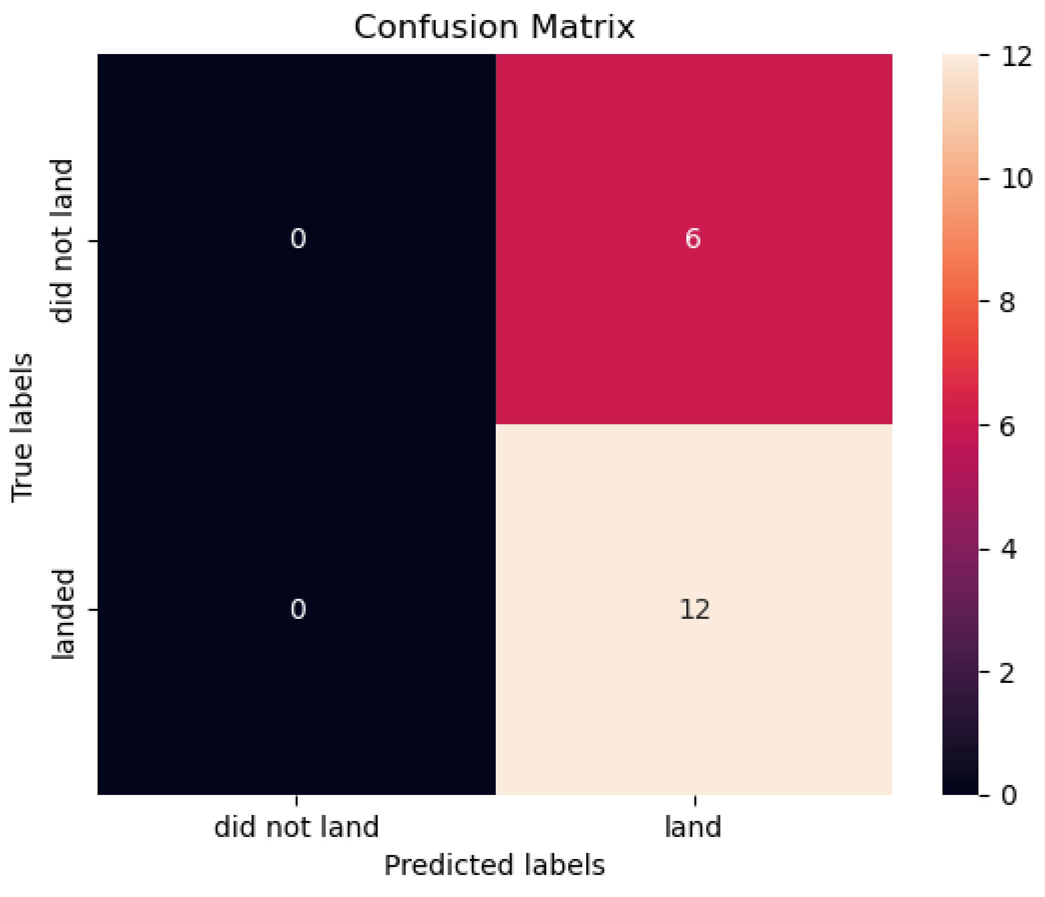
Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy
- Decision Tree was slightly better 86 percent



Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation
- The model has a moderate accuracy of 67%.The model is particularly good at predicting "landed" instances (no false negatives), but it has a high false positive rate, meaning it often predicts "landed" when the true label is "did not land."This might indicate that the model is overly sensitive or prone to false alarms.



Conclusions

- Point 1
- Point 2
- Point 3
- Point 4
- …

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!