IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Jean Shepherd>
<18 Dec, 2024>

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
    - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch
- Problems you want to find answers
    - Predict if the first stage will lang

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Use API call with *api.spacexdata.com*
  - Use Web Scraping with Python library *beautifulsoup4*
- Perform data wrangling
  - Use *pandas* and *numpy* process data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Use *scikit-learn* to build, tune, evaluate classification models

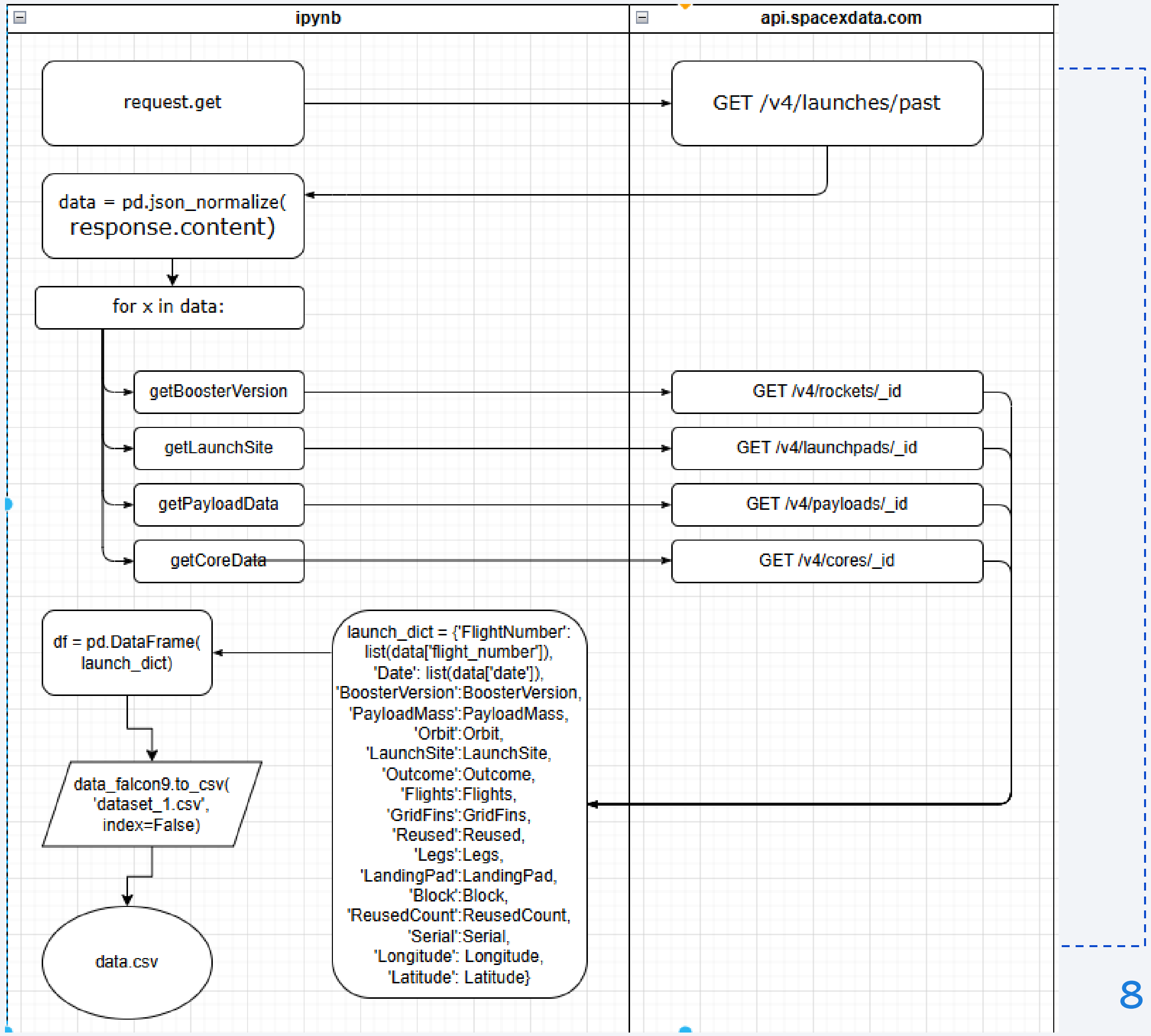
Data Collection

- Describe how data sets were collected.
  - Method 1: web scraping
  - Method 2: SpaceX API call
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

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

- https://github.com/JeanG00/data-analysis-space-x/blob/master/notebooks/apply_1_SpaceX_api.ipynb

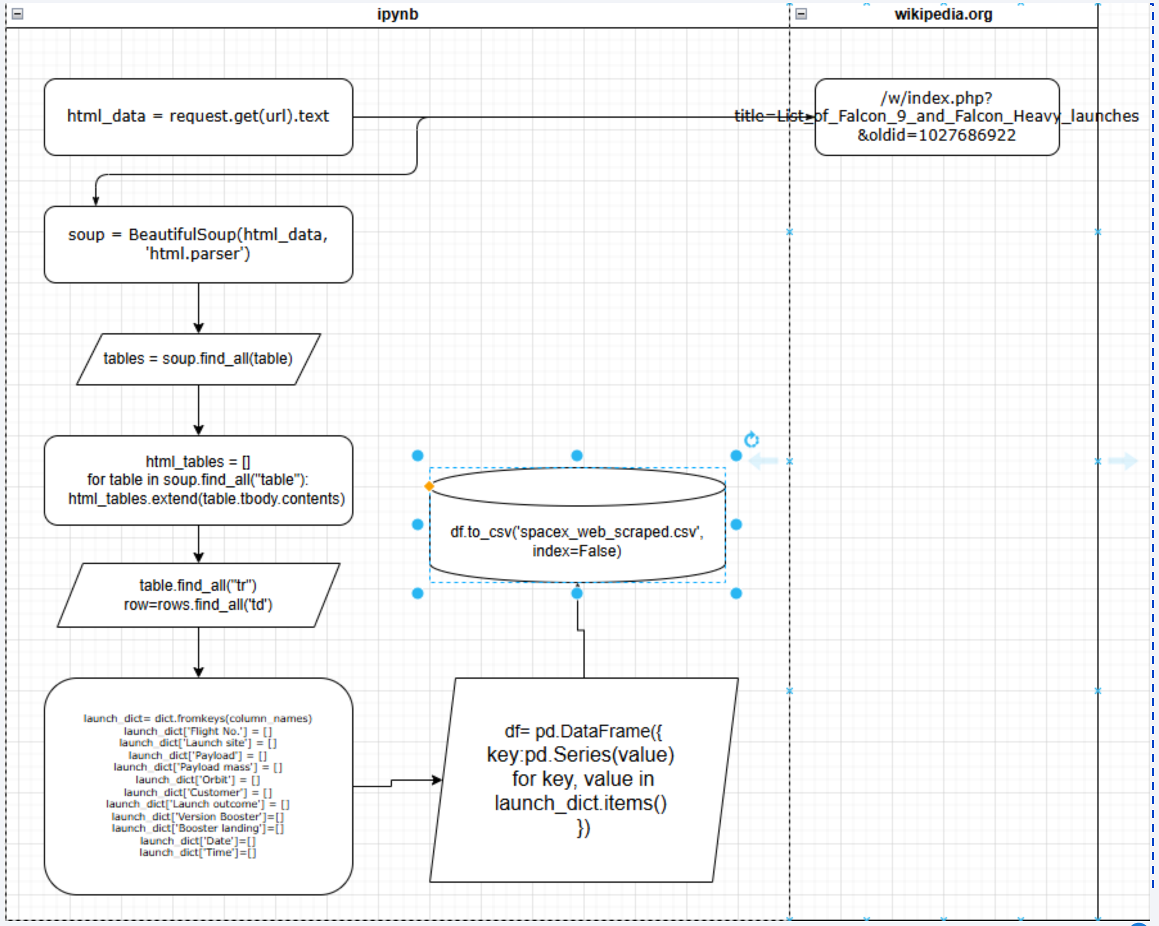


```
launch_dict = {'FlightNumber':
list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- https://github.com/JeanG00/data-analysis-space-x/blob/master/notebooks/apply_1B_SpaceX_scraping.ipynb



```
/w/index.php?
title=List_of_Falcon_9_and_Falcon_Heavy_launches
&oldid=1027686922
```

```
html_data = request.get(url).text
```

```
soup = BeautifulSoup(html_data,
'html.parser')
```

```
tables = soup.find_all(table)
```

```
html_tables = []
for table in soup.find_all("table"):
html_tables.extend(table.tbody.contents)
```

```
table.find_all("tr")
row=rows.find_all('td')
```

```
launch_dict= dict.fromkeys(column_names)
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
df= pd.DataFrame({
key:pd.Series(value)
for key, value in
launch_dict.items()
})
```

```
df.to_csv('spacex_web_scraped.csv',
index=False)
```

Data Wrangling

- Describe how data were processed
  - String manipulation with data cells to remove unnecessary parts
  - Identify and calculate the percentage of the missing values in each attribute
  - Identify which columns are numerical and categorical
- key phrases
  - df.isnull().sum()
  - len(df)
  - df.loc[]
  - df.iloc[]

- https://github.com/JeanG00/data-analysis-space-x/blob/master/notebooks/apply_1C_SpaceX_data_wrangling.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
  - Line plot - best suited for trend-based visualizations of data over a period of time
  - Scatter plot - a useful way of comparing variables against each other
  - Bar plot - a way of representing data where the *length* of the bars represents the magnitude/size of the feature/variable
- https://github.com/JeanG00/data-analysis-space-x/blob/master/notebooks/apply_2B_EDA.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - DROP TABLE
  - CREATE TABLE
  - SLECT * FROM TABLE GROUP BY COLUMN
  - SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Booster_Version" LIKE "%F9 v1.1%"
  - SELECT COUNT(*) as "Mission", "Mission_Outcome" FROM SPACEXTBL GROUP BY "Mission_Outcome"
- https://github.com/JeanG00/data-analysis-space-x/blob/master/notebooks/apply_2_SQL.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
  - Folium.Circle - drawing circle overlays on a map
  - Folium.map.Marker - Create a simple marker on the map, with optional popup text
  - Folium.Icon - enrich user experience
  - Folium.Popup - making the web page more interactive
  - Folium.PolyLine - creates line object overlays on the map
- https://github.com/JeanG00/data-analysis-space-x/blob/master/notebooks/apply_3_folium_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

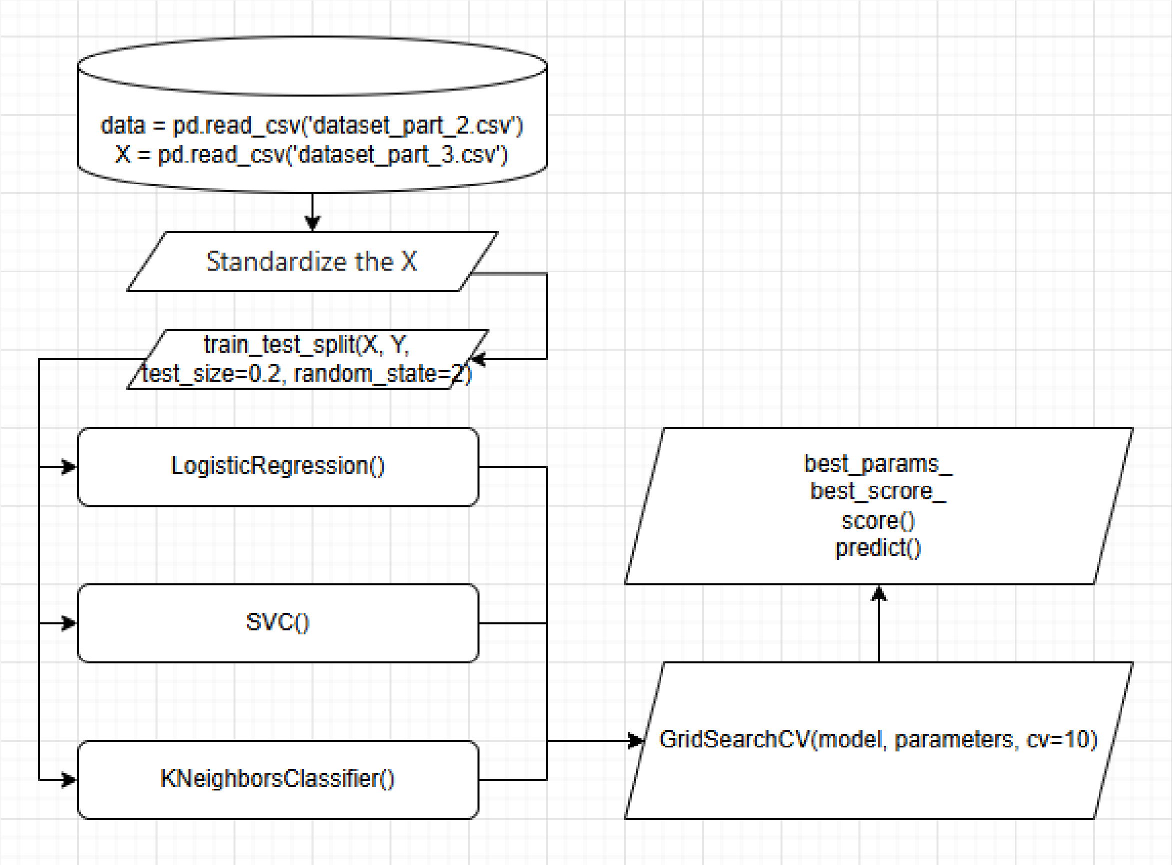
- Summarize what plots/graphs and interactions you have added to a dashboard
  - Pie chart - A. to see which sites has the largest success count B. to check detailed success rate (class=0 vs. class=1) on one site
  - Scatter chart - visually observe how payload may be correlated with mission outcomes for selected site(s)
- https://github.com/JeanG00/data-analysis-space-x/blob/master/notebooks/apply_3B_plotly_dash.py

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model

    1. Load data
    2. Standardize the data
    3. Create train and test data sets
    4. Create different models and calc the score
    5. Use GridSearchCV find best model

- https://github.com/JeanG00/data-analysis-space-x/blob/master/notebooks/apply_4_SpaceX_MLP.ipynb



data = pd.read_csv('dataset_part_2.csv')
X = pd.read_csv('dataset_part_3.csv')

Standardize the X

train_test_split(X, Y, test_size=0.2, random_state=2)

LogisticRegression()

SVC()

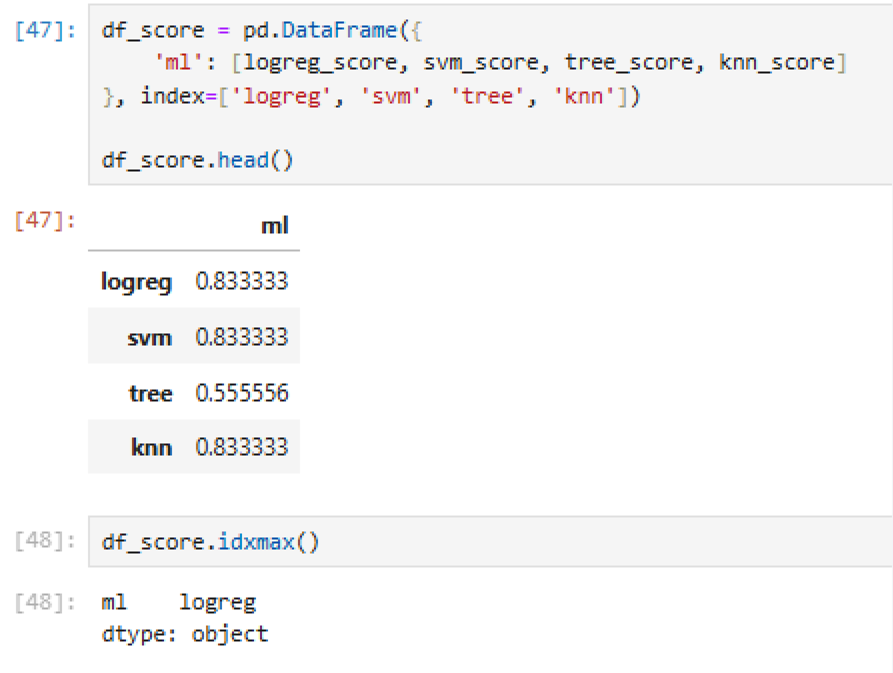
KNeighborsClassifier()

GridSearchCV(model, parameters, cv=10)

best_params_
best_scrore_
score()
predict()

Results

- Exploratory data analysis results
  - Models share similar scores
- Interactive analytics demo in screenshots
  - As picture right displays
- Predictive analysis results
  - Predictions share similar results



[47]:

```
df_score = pd.DataFrame({
    'ml': [logreg_score, svm_score, tree_score, knn_score]
}, index=['logreg', 'svm', 'tree', 'knn'])

df_score.head()
```

[47]:

	ml
logreg	0.833333
svm	0.833333
tree	0.555556
knn	0.833333

[48]:

```
df_score.idxmax()
```

[48]:

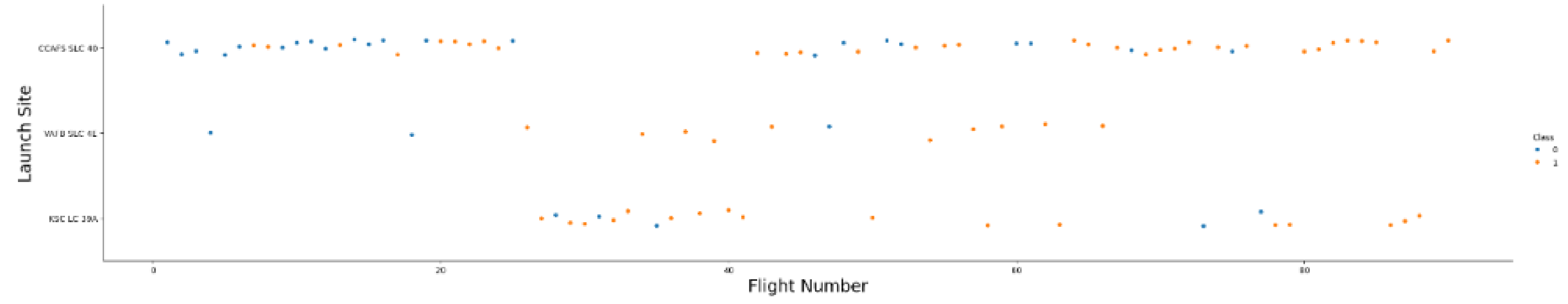
```
ml    logreg
dtype: object
```

Section 2

Insights drawn from EDA

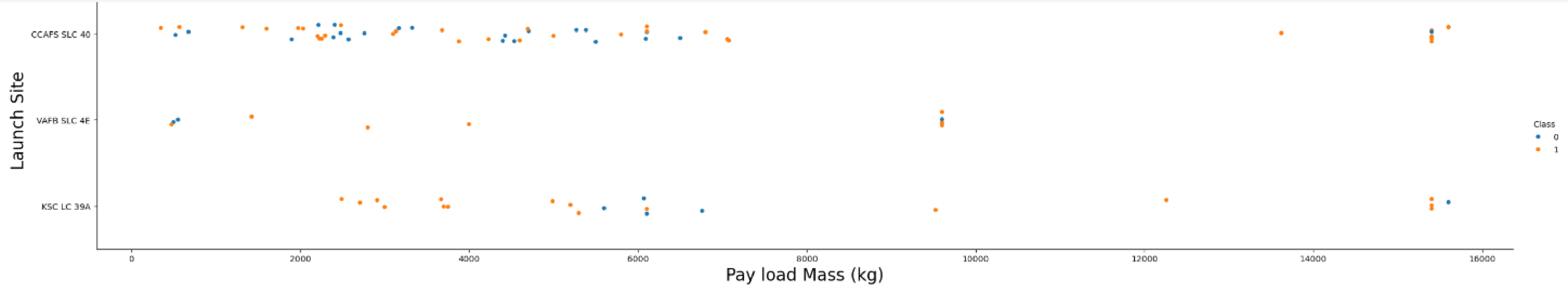
Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations
  - ○ Most rockets was launched at CCAFS SLC 40
  - ○ KSC LC 39A was put into use after 25 launches
  - ○ VAFB SLC 4E does not rarely launch rockets



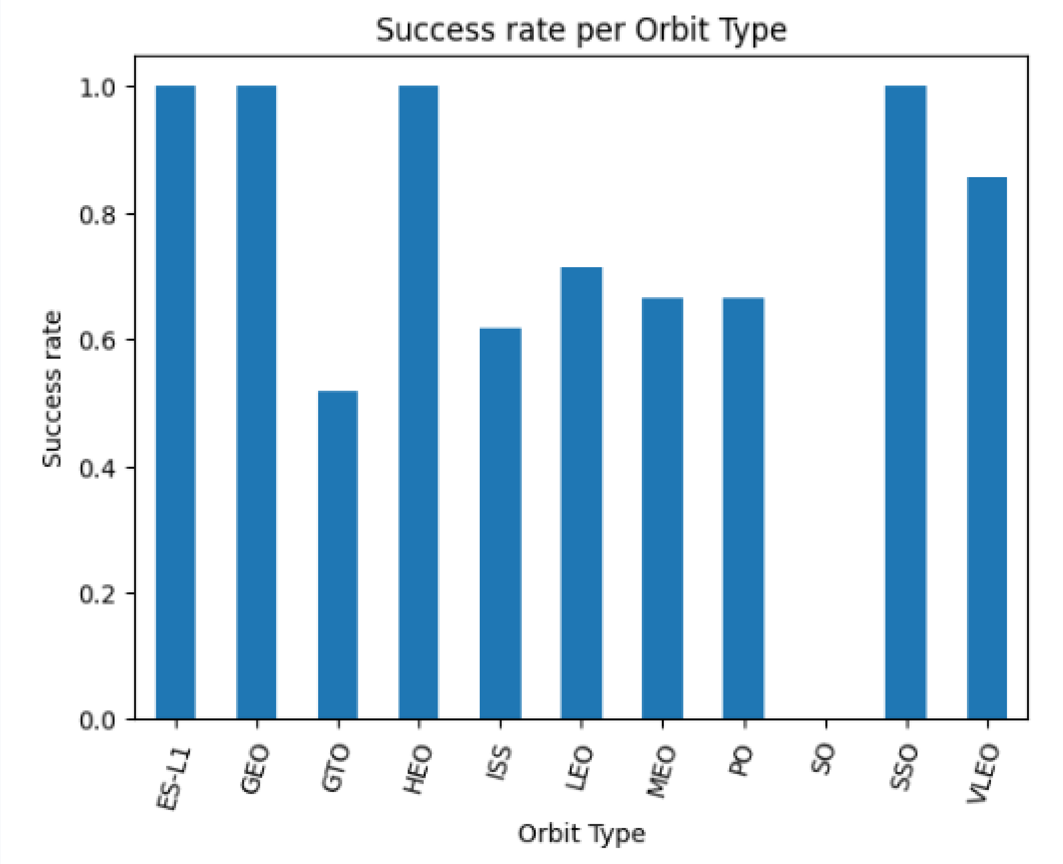
Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- Show the screenshot of the scatter plot with explanations
  - ○ CCAFS SLC 40 and KSC LC 39A was used for heavy-weight rockets launch
  - ○ VAFB SLC 4E was used for middle-weight rockets launch



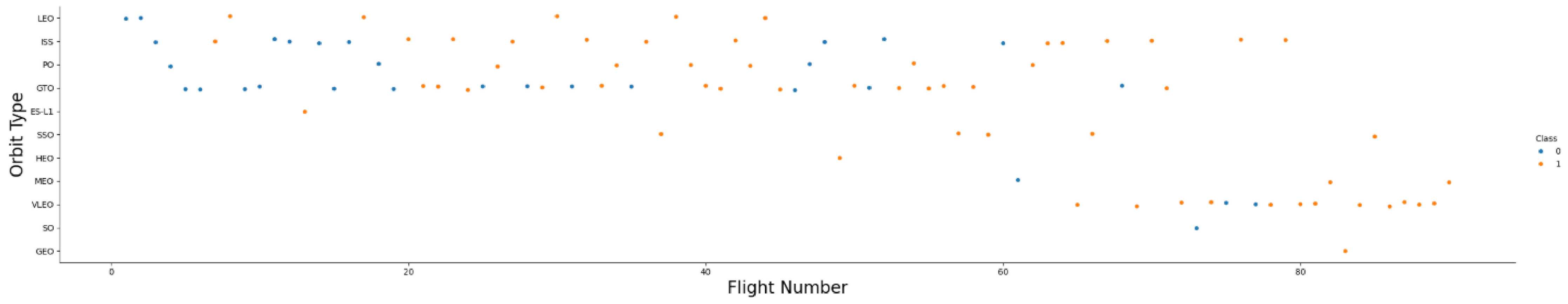
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations
    - Orbit type with GTO has lowest success rate
    - Orbit type with ISS, MEO, PO share similar success rate
    - Orbit type with ES-L1, GEO, HEO, SSO has remarkable success rate



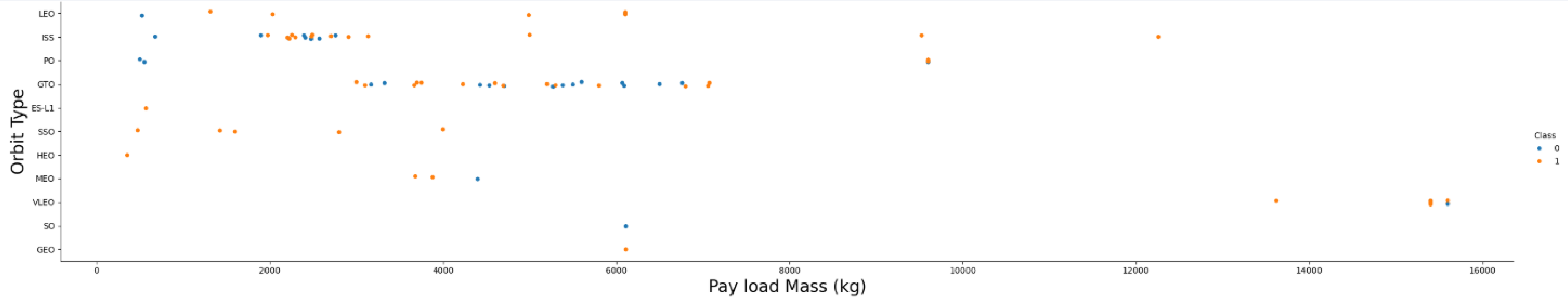
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations
  - ○ Rockets were launched at lower orbit in the beginning, as success rate rises, they goes to higher orbit.
  - ○ Orbit type with LEO, ISS, FO, GTO were frequently launched



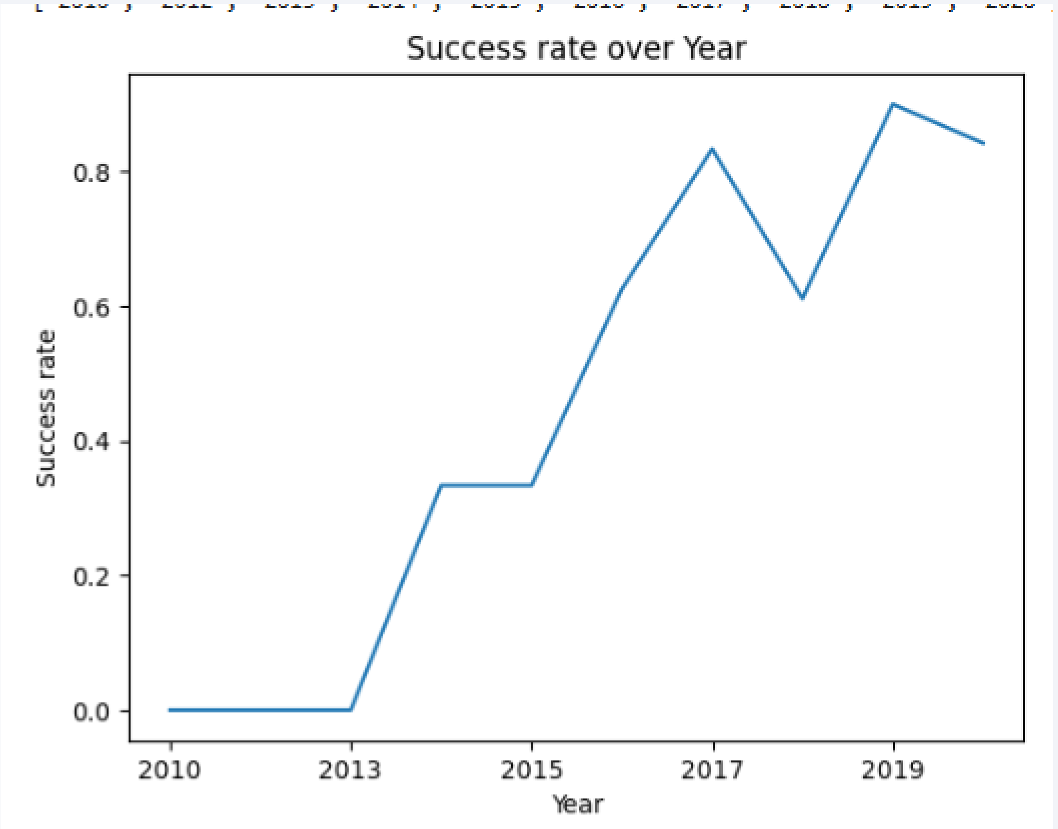
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations
  - ○ Rockets launch more light-weight payload than heavy-weight
  - ○ Heavy-weight above 15,000 kg were only launched at VLEO for observation and test purpose



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations
  - The first success launch starts from 2013
  - The success rates goes up quickly as year goes by
  - The success rates drops between 2017 and 2018, maybe new technical test carried out?



All Launch Site Names

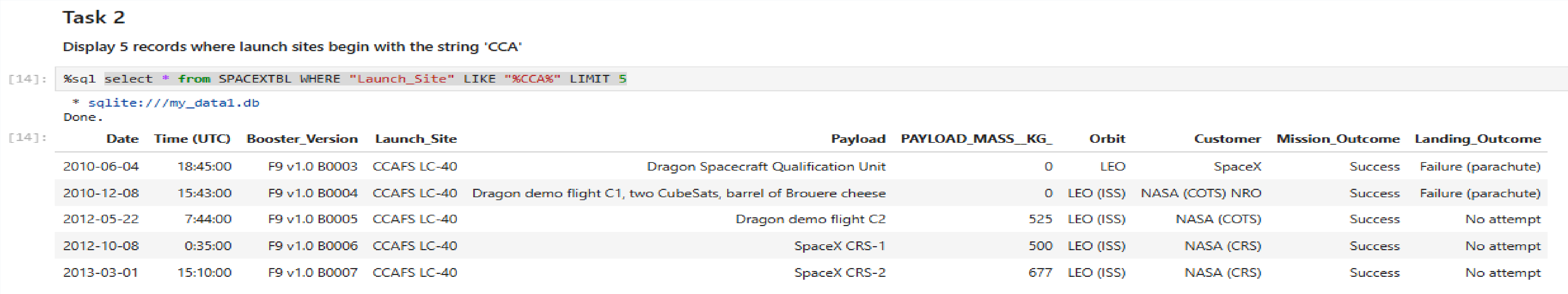
- Find the names of the unique launch sites
  - SELECT DISTINCT Launch_Site FROM SPACEXTBL

- Present your query result with a short explanation here
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E

  - There are 4 unique launch sites, CCAFS LC-40 and CCAFS SLC-40 are probably close to each other

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
    - SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE "%CCA%" LIMIT 5
- Present your query result with a short explanation here
    - From the booster version number we can see CCA% sites are put into use quite early



Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
[14]: %sql select * from SPACEXTBL WHERE "Launch_Site" LIKE "%CCA%" LIMIT 5
```

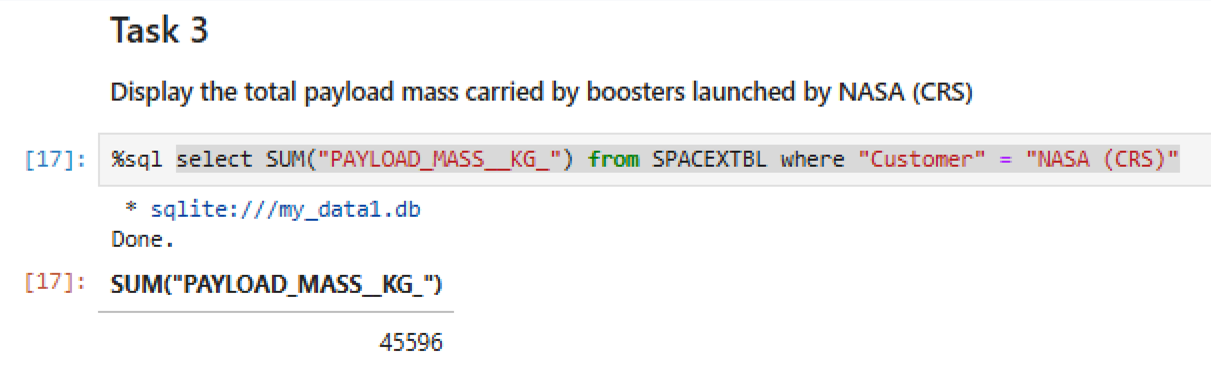
```
 * sqlite:///my_data1.db
Done.
```

[14]:

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
  - SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHER "Customer" = "NASA (CRS)"
- Present your query result with a short explanation here
  - `WHERE` clause are used for filtering
  - `SUM` clause are used for math calculation



Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
[17]: %sql select SUM("PAYLOAD_MASS__KG_") from SPACEXTBL where "Customer" = "NASA (CRS)"
```

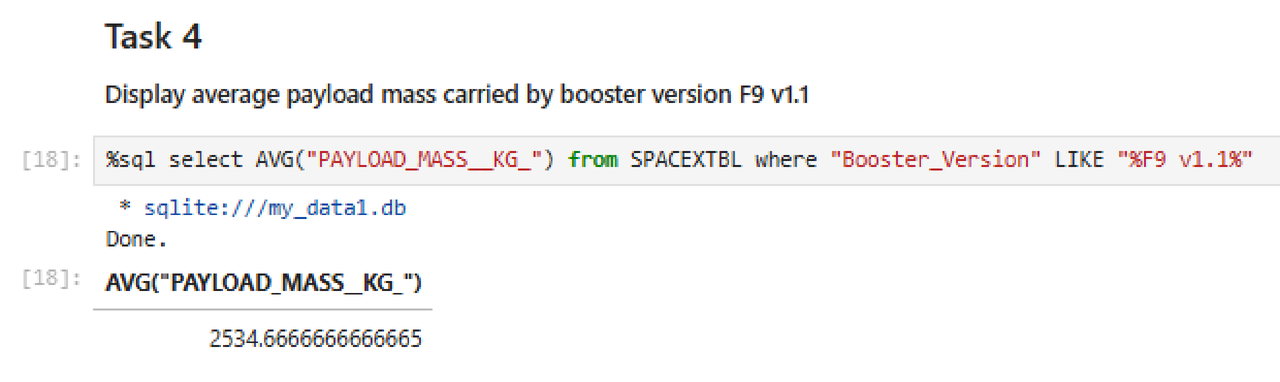
```
 * sqlite:///my_data1.db
Done.
```

[17]:

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
  - SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Booster_Version" LIKE "%F9 v1.1%"
- Present your query result with a short explanation here
  - The average payload mass in kilogram is 2535(kg)



Task 4

Display average payload mass carried by booster version F9 v1.1

```
[18]: %sql select AVG("PAYLOAD_MASS__KG_") from SPACEXTBL where "Booster_Version" LIKE "%F9 v1.1%"
```

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

[18]:

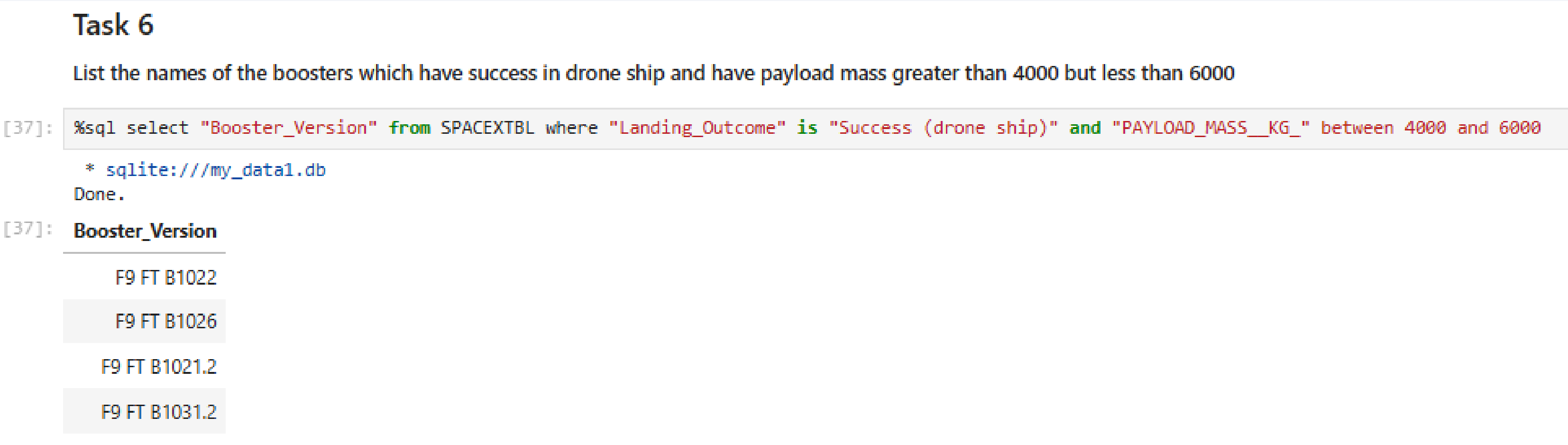
AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
  - SELECT MIN("Date") FROM SPACEXTBL WHERE "Landing_Outcome" = "Success (ground pad)" AND "Mission_Outcome" = "Success"
  - Present your query result with a short explanation here
    - By `MIN()` function we get 2015-12-22

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
  - select "Booster_Version" from SPACEXTBL where "Landing_Outcome" is "Success (drone ship)" and "PAYLOAD_MASS__KG_" between 4000 and 6000
- Present your query result with a short explanation here
  - Falcon 9



Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
[37]: %sql select "Booster_Version" from SPACEXTBL where "Landing_Outcome" is "Success (drone ship)" and "PAYLOAD_MASS__KG_" between 4000 and 6000
```

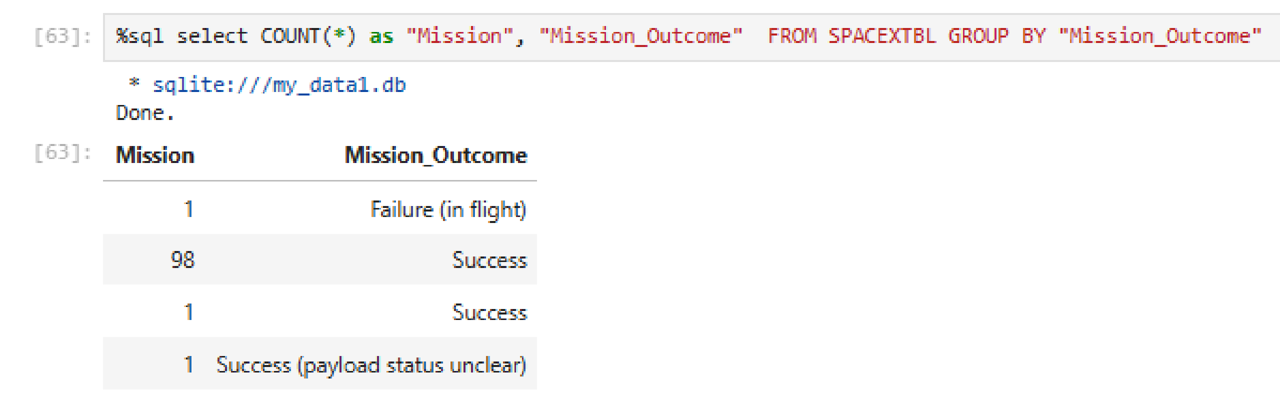
```
 * sqlite:///my_data1.db
Done.
```

[37]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
  - select COUNT(*) as "Mission", "Mission_Outcome" FROM SPACEXTBL GROUP BY "Mission_Outcome"
- Present your query result with a short explanation here
  - We can see that success rate are quite high



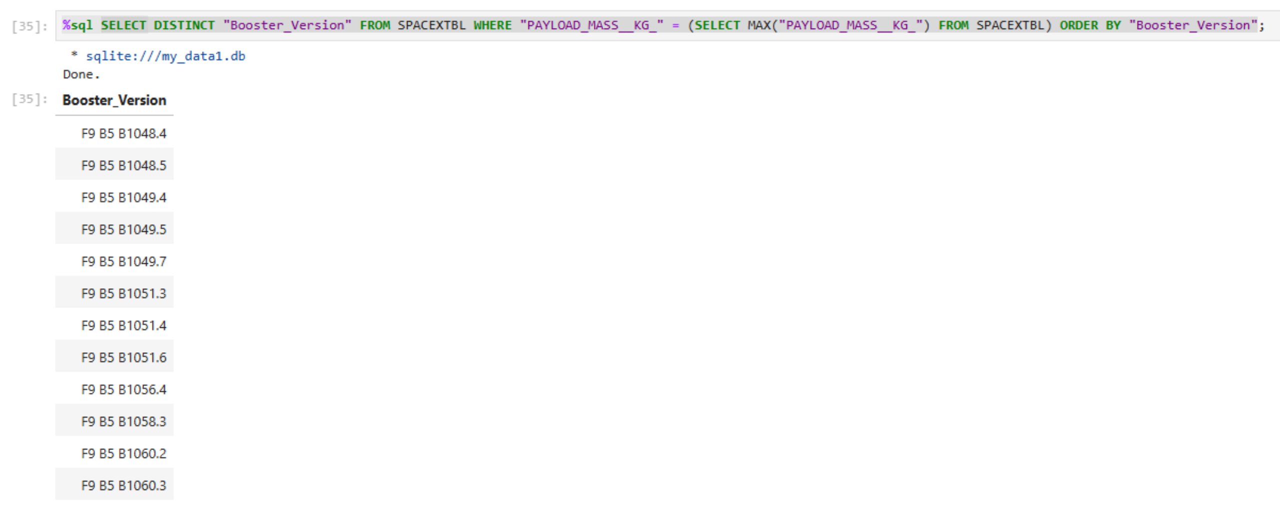
```
[63]: %sql select COUNT(*) as "Mission", "Mission_Outcome"   FROM SPACEXTBL GROUP BY "Mission_Outcome"

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

[63]:

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
  - SELECT DISTINCT "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL) ORDER BY "Booster_Version"
- Present your query result with a short explanation here
  - Falcon 9 B5



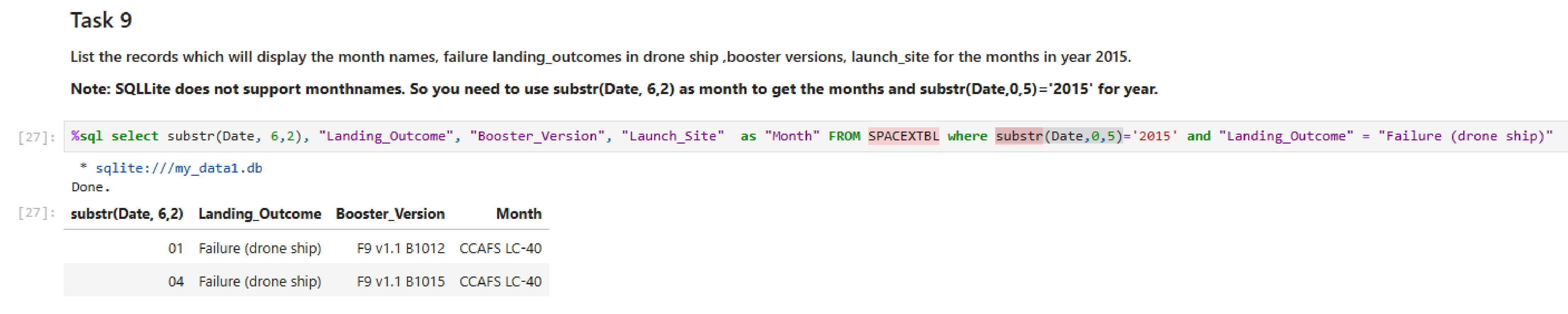
```
[35]: %sql SELECT DISTINCT "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL) ORDER BY "Booster_Version";

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

[35]:

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - select substr(Date, 6,2), "Landing_Outcome", "Booster_Version", "Launch_Site" as "Month" FROM SPACEXTBL where substr(Date,0,5)='2015' and "Landing_Outcome" = "Failure (drone ship)"
- Present your query result with a short explanation here



Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
[27]: %sql select substr(Date, 6,2), "Landing_Outcome", "Booster_Version", "Launch_Site"  as "Month" FROM SPACEXTBL where substr(Date,0,5)='2015' and "Landing_Outcome" = "Failure (drone ship)"
```

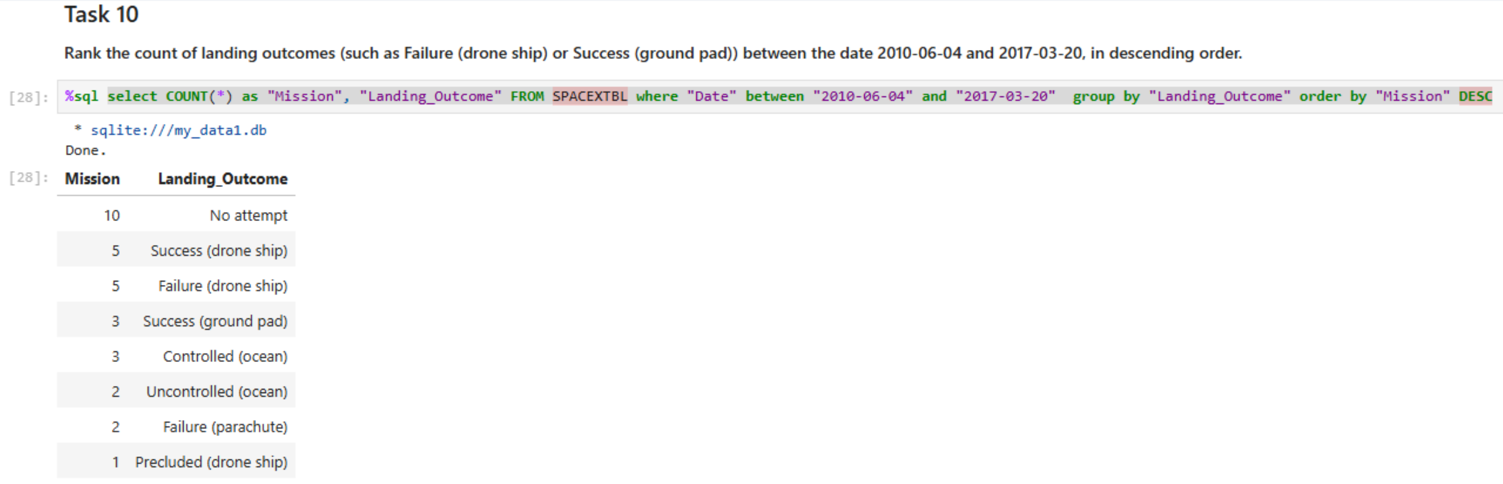
```
 * sqlite:///my_data1.db
Done.
```

[27]:

substr(Date, 6,2)	Landing_Outcome	Booster_Version	Month
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

- 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
  - select COUNT(*) as "Mission", "Landing_Outcome" FROM SPACEXTBL where "Date" between "2010-06-04" and "2017-03-20" group by "Landing_Outcome" order by "Mission" DESC
- Present your query result with a short explanation here
  - Landing Outcomes with "No attempt" is the first outcome



Task 10

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.

```
[28]: %sql select COUNT(*) as "Mission", "Landing_Outcome" FROM SPACEXTBL where "Date" between "2010-06-04" and "2017-03-20"  group by "Landing_Outcome" order by "Mission" DESC
```

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

[28]:

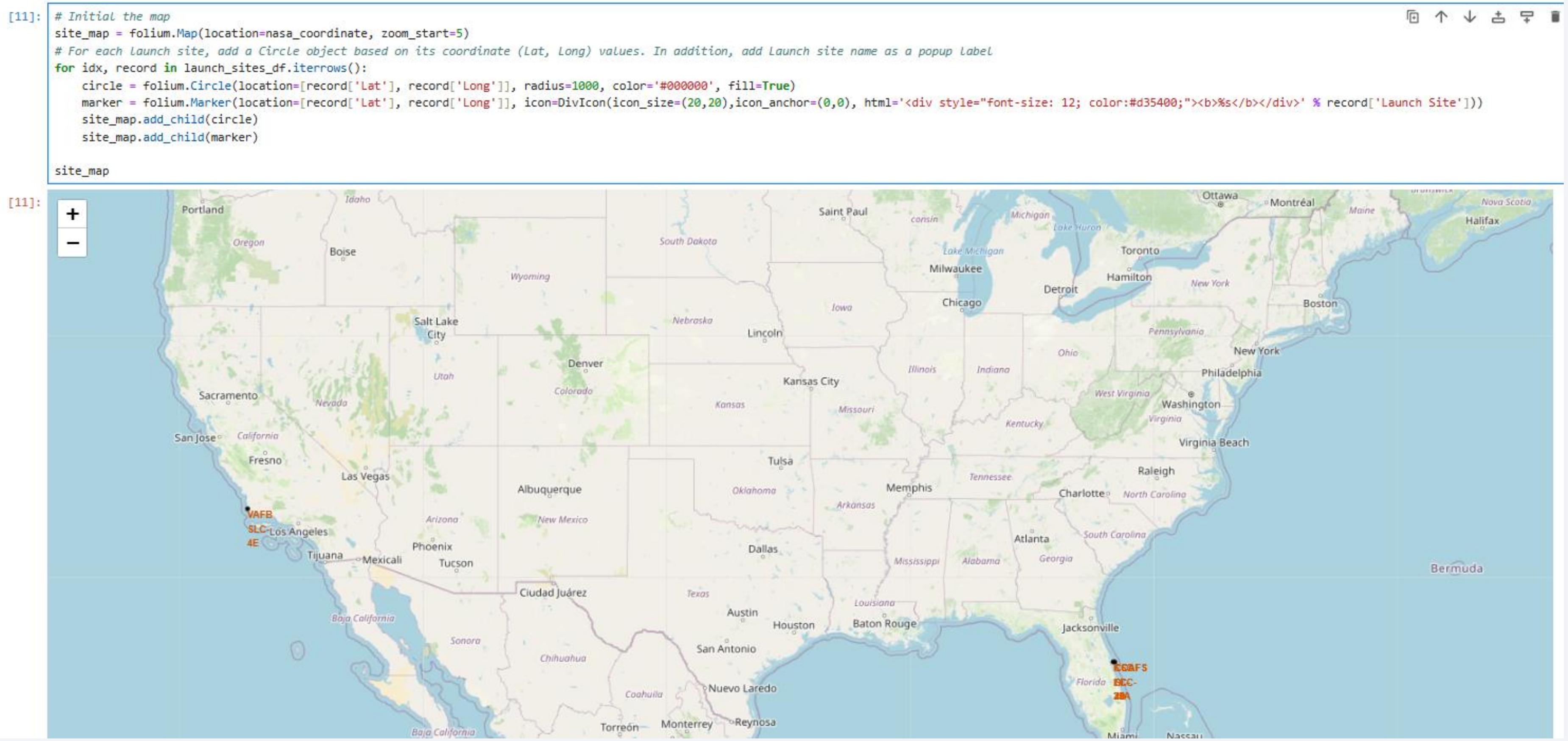
Mission	Landing_Outcome
10	No attempt
5	Success (drone ship)
5	Failure (drone ship)
3	Success (ground pad)
3	Controlled (ocean)
2	Uncontrolled (ocean)
2	Failure (parachute)
1	Precluded (drone ship)

Section 3

Launch Sites Proximities Analysis

Launch site distribution

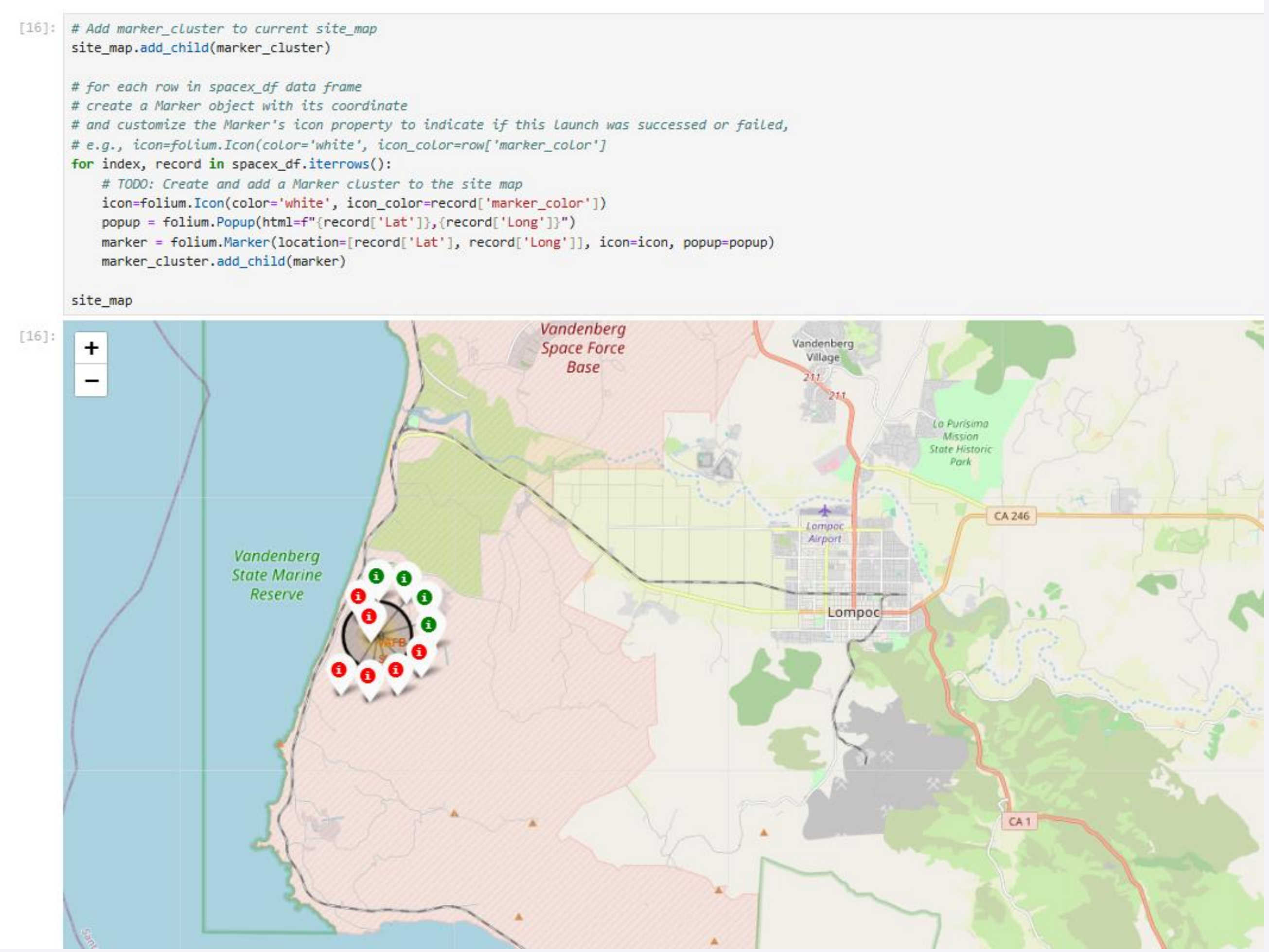
- Explain the important elements and findings on the screenshot
  - There are two major launch sites, one near to Pacific Ocean and the other to Atlantic Ocean



```
[11]: # Initial the map
      site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
      # For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label
      for idx, record in launch_sites_df.iterrows():
          circle = folium.Circle(location=[record['Lat'], record['Long']], radius=1000, color='#000000', fill=True)
          marker = folium.Marker(location=[record['Lat'], record['Long']], icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0), html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % record['Launch Site']))
          site_map.add_child(circle)
          site_map.add_child(marker)

      site_map
[11]:
```

Success/failed launches for each site

- Explain the important elements and findings on the screenshot
  - The coordinate data are quite close to each other

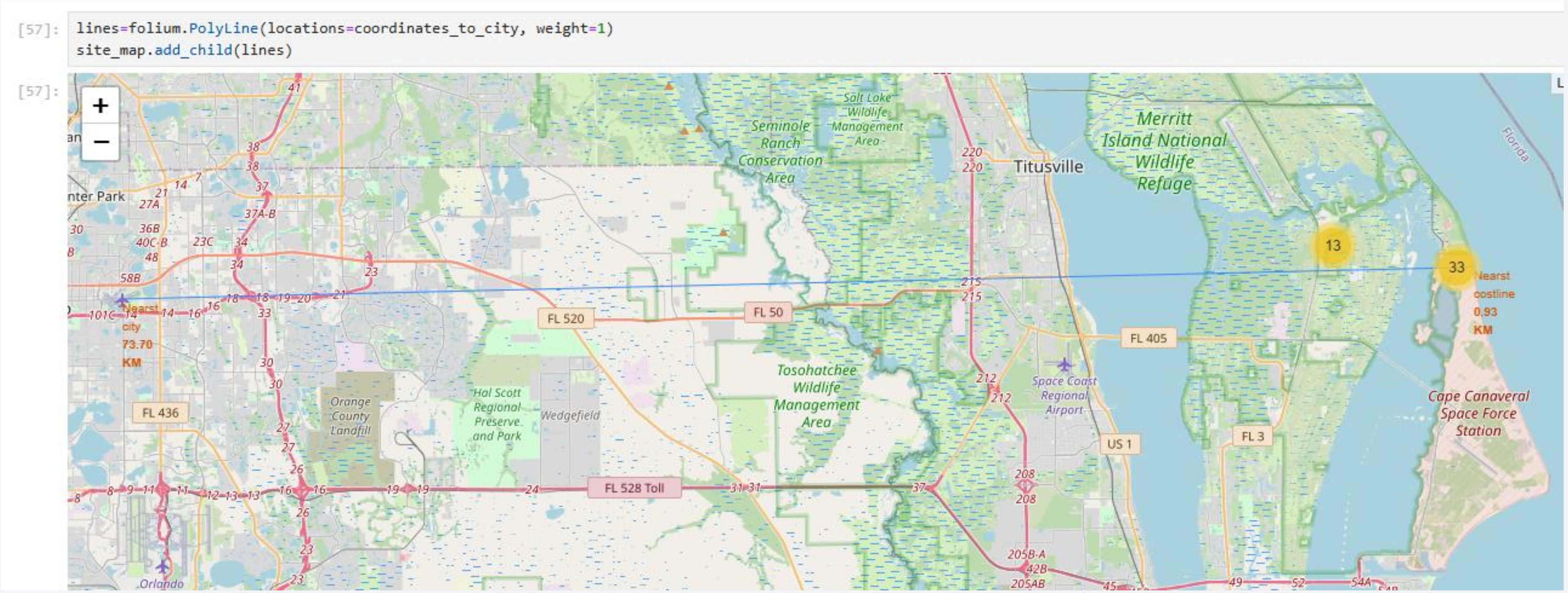


```
[16]: # Add marker_cluster to current site_map
      site_map.add_child(marker_cluster)

      # 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():
          # TODO: Create and add a Marker cluster to the site map
          icon=folium.Icon(color='white', icon_color=record['marker_color'])
          popup = folium.Popup(html=f"{record['Lat']},{record['Long']}")
          marker = folium.Marker(location=[record['Lat'], record['Long']], icon=icon, popup=popup)
          marker_cluster.add_child(marker)

      site_map
```

[16]:
+
−
Vandenberg Space Force Base
Vandenberg Village
211
La Purisima Mission State Historic Park
Lompoc Airport
CA 246
Vandenberg State Marine Reserve
Lompoc
CA 1

Calculate the distances between a launch site to its proximities

- Explain the important elements and findings on the screenshot
  - CCAFS LC-40 to Nearest costline 0.93km
  - CCAFS LC-40 to Nearest city (Orlando Airport) 73.70km

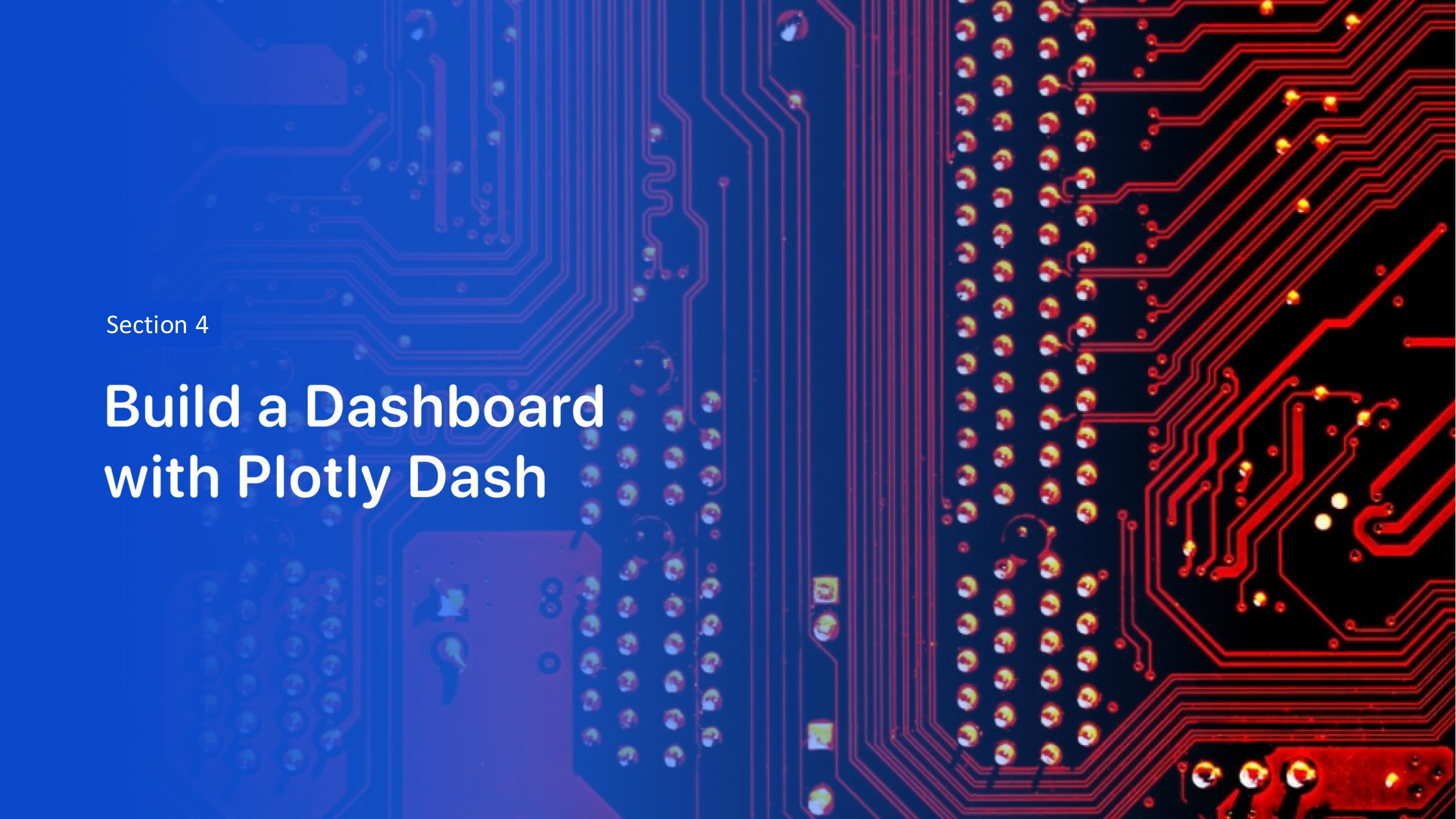


```
[57]: lines=folium.PolyLine(locations=coordinates_to_city, weight=1)
      site_map.add_child(lines)
```

[57]:
+
−
13
33
Nearst
costline
0.93
KM
Nearst
city
73.70
KM
Titusville
Merritt Island National Wildlife Refuge
Seminole Ranch Conservation Area
Salt Lake Wildlife Management Area
Tosohatchee Wildlife Management Area
Space Coast Regional Airport
Cape Canaveral Space Force Station
Hal Scott Regional Preserve and Park
Wedgefield
Orange County Landfill
Orlando
Florida
FL 520
FL 50
FL 405
FL 436
FL 528 Toll
US 1
FL 3

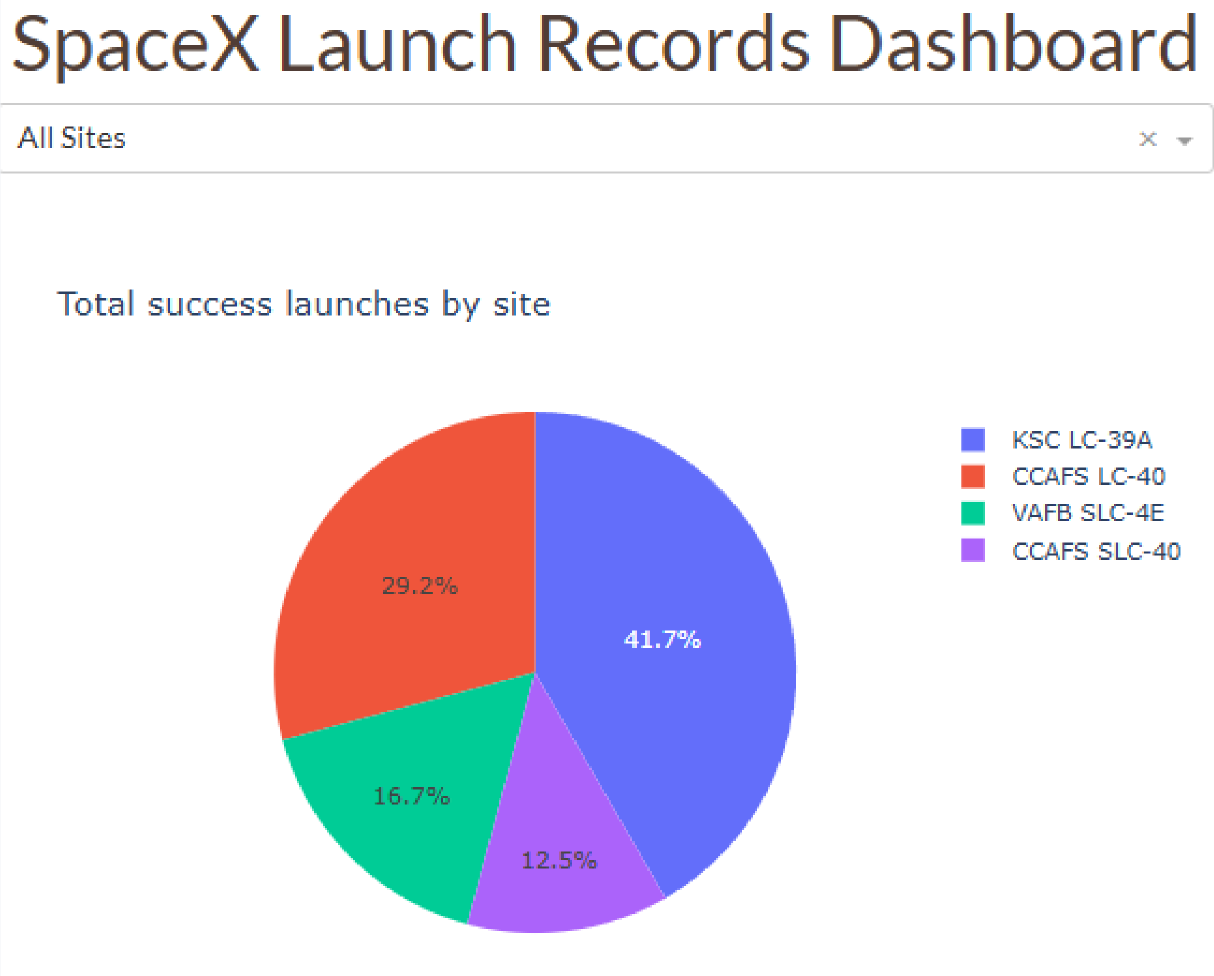
Section 4

Build a Dashboard with Plotly Dash



Total success launches by site

- Explain the important elements and findings on the screenshot
  - KSC LC-39A has highest success rate of launches
  - CCAFS SLC-40 has lowest success rate of launches

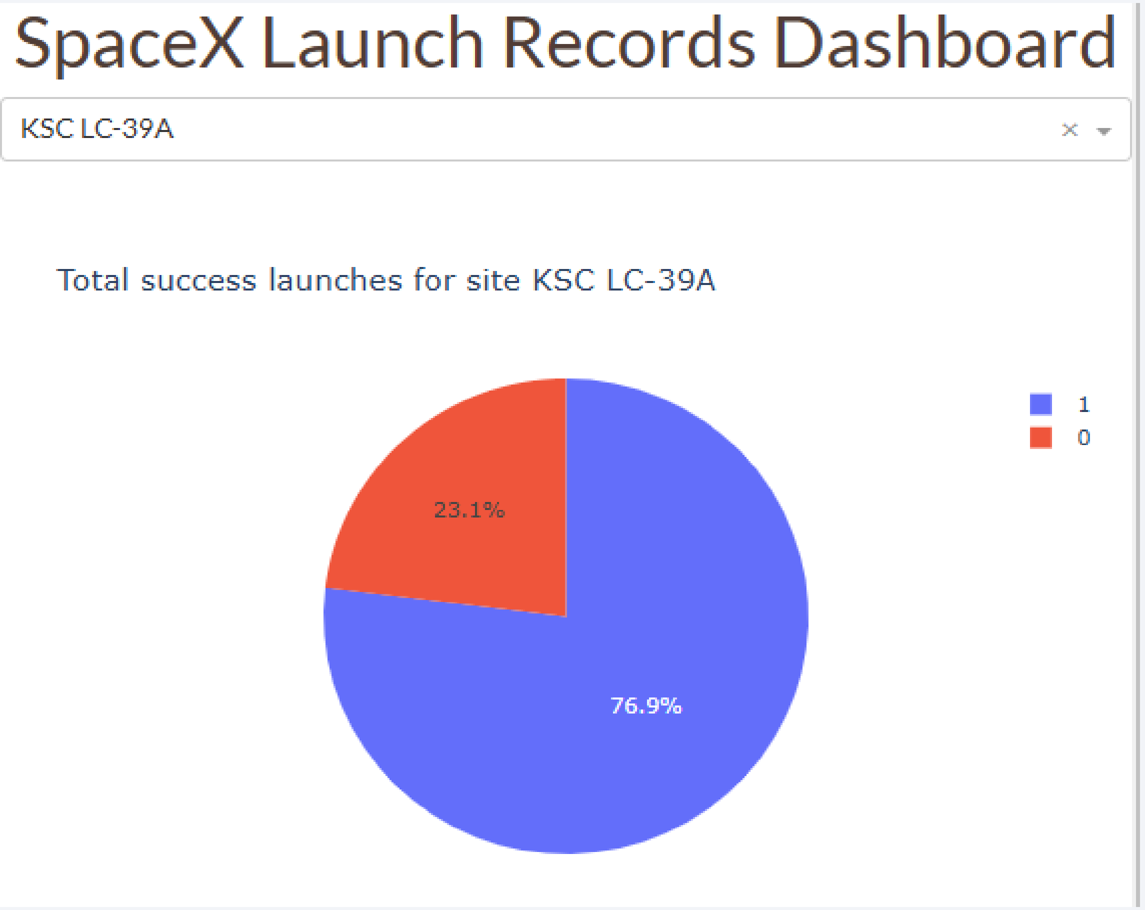


SpaceX Launch Records Dashboard

All Sites

Site of highest launch success ratio

- Explain the important elements and findings on the screenshot
  - The success rate is approximately 76.9%



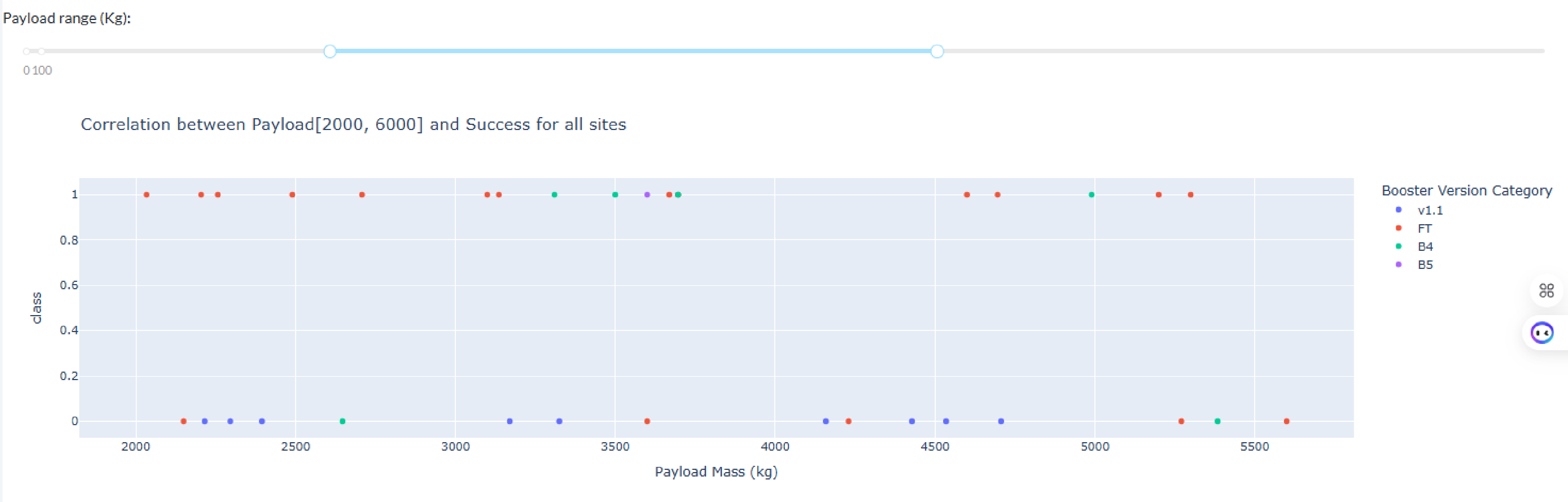
SpaceX Launch Records Dashboard

KSC LC-39A

Total success launches for site KSC LC-39A

Payload vs. Launch Outcome slider

- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.
  - Payload range between 5500kg and 9000kg has the largest success rate of 1



Payload range (Kg):

Correlation between Payload[2000, 6000] and Success for all sites

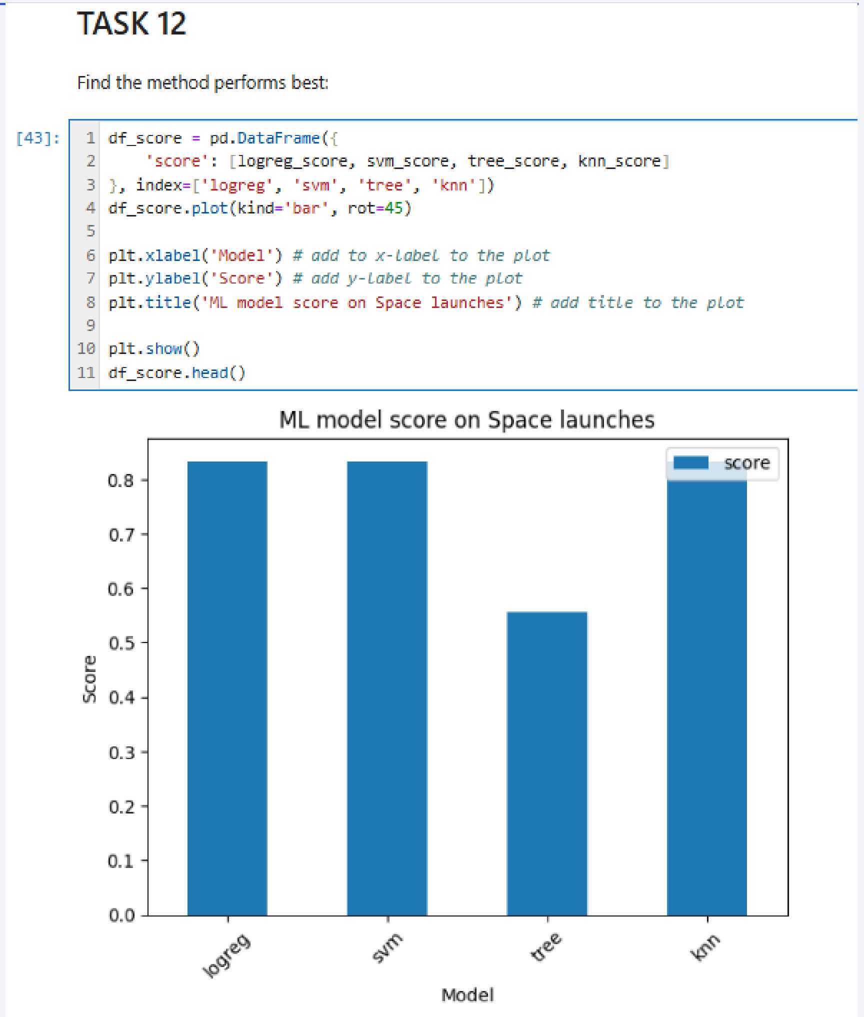
Section 5

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
    - ○ Models share similar accuracy
    - ○ Probable cause: dataset too small



TASK 12

Find the method performs best:

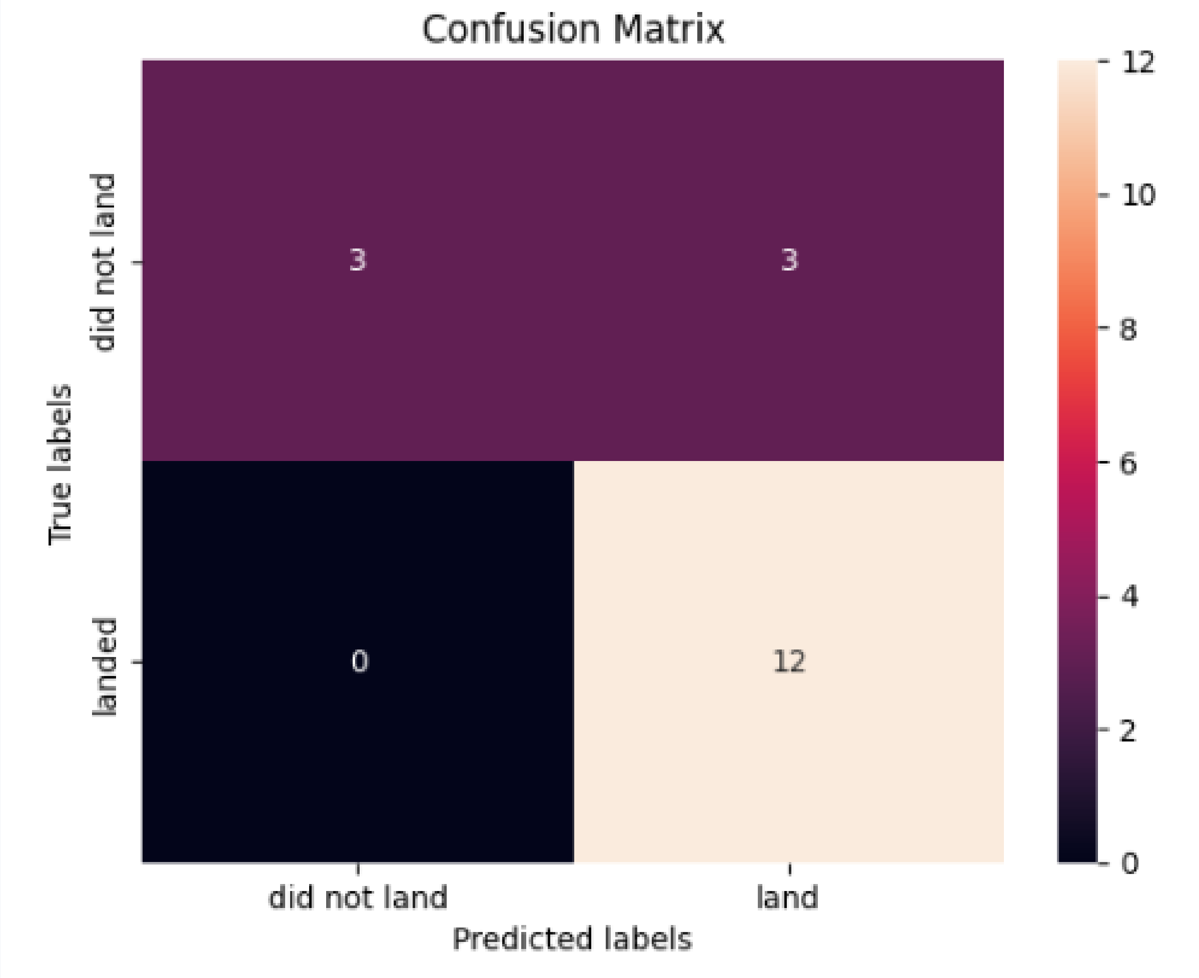
[43]:

```
df_score = pd.DataFrame({
    'score': [logreg_score, svm_score, tree_score, knn_score]
}, index=['logreg', 'svm', 'tree', 'knn'])
df_score.plot(kind='bar', rot=45)

plt.xlabel('Model') # add to x-label to the plot
plt.ylabel('Score') # add y-label to the plot
plt.title('ML model score on Space launches') # add title to the plot

plt.show()
df_score.head()
```

ML model score on Space launches

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation
  - A true positive is an outcome where the model correctly predicts the positive class.
  - A false positive is an outcome where the model incorrectly predicts the positive class.



Conclusions

- Dataset size affect predict accuracy significantly
- Good classifier model parameters will accelerate the process finding best models
- Good usage of appropriate chart type helps us understanding data easily
- Data analysis is crucial in various fields such as business, science, research, and many others to make informed decisions and predictions based on data patterns and trends

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
- df_score.idxmax()
- df = data._get_numeric_data()
- popup = folium.Popup(html=f"{record['Lat']},{record['Long']}")
- df['Landing_Outcome'].unique()
- select MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL
- select DISTINCT Launch_Site FROM SPACEXTBL
- colunm_name.replace(r'.', '').replace(r'(', '').replace(r')', '').strip()

Thank you!