IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jamie Bradsher
6/26/2022

Outline

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

Executive Summary

Summary of methodologies

1. Business understanding
2. Data understanding
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment

- Summary of all results:

Machine learning analysis shows after selecting the best hyperparameters for the decision tree classifier using the validation data, 83.33% accuracy was achieved on the test data.

Introduction

- Project background and context

A data scientist working for a new rocket company the job is to determine the price of each launch. This will be done by gathering data about Space X to predict outcome for Space Y and help determine cost. In order to do this data will have to be collected, wrangled, visualized and a model will need to be created.

- Problems you want to find answers

Determine if SpaceX able to reuse the first stage.

Determine if the first stage will land and then determine the cost of a launch.

Training a machine learning model to predict if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  Use data from SpaceX by using API REST to collect key data

- Perform data wrangling
  - Collected data in data frame is checked for Nulls and checked for shape of data frame
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build the model – split data, conduct fit and get score to determine which is closest to 1

Data Collection

- Describe how data sets were collected.

  SpaceX launch data was collected from SpaceX REST API endpoint

- Data collection process use key phrases and flowcharts

  Process for data collection steps will show a process which validates end result findings

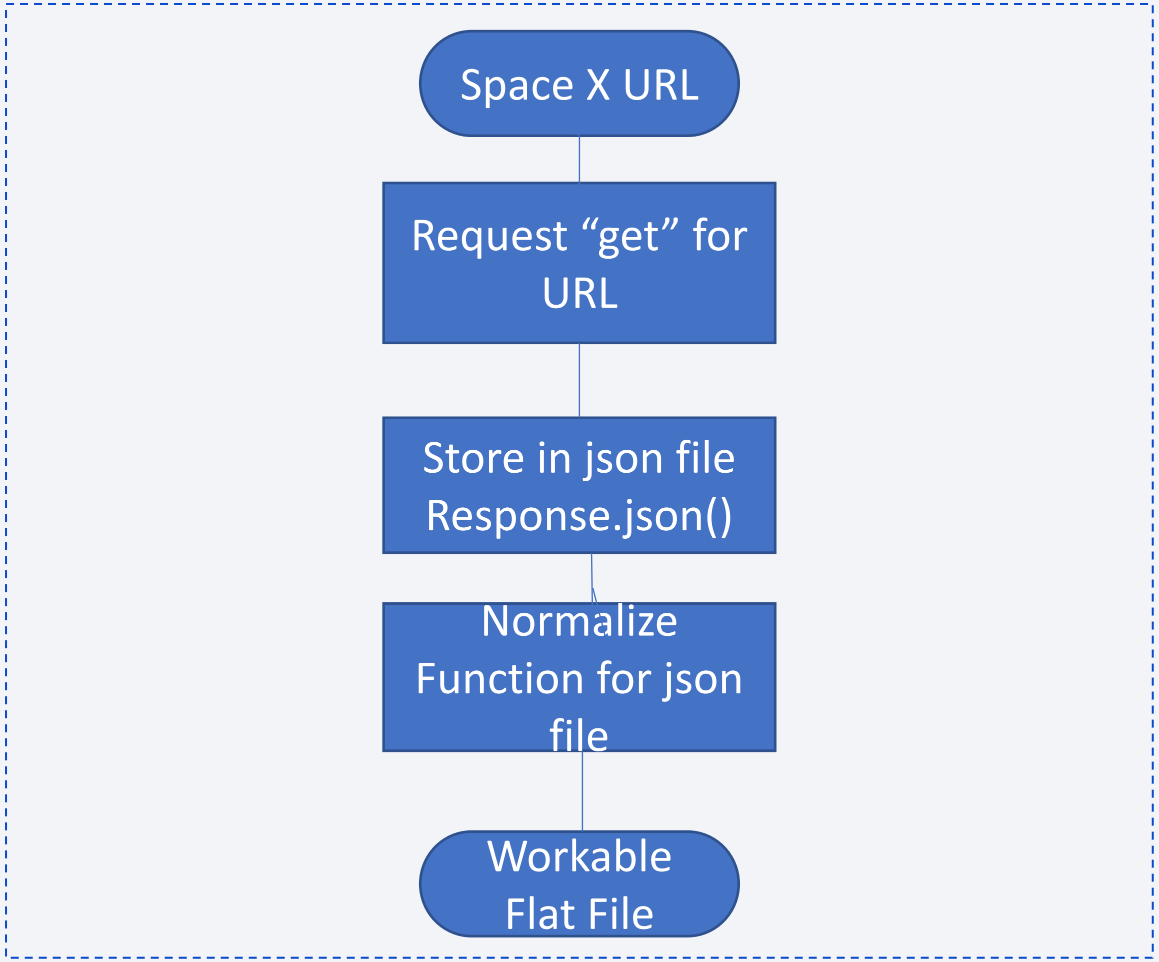
Data Collection – SpaceX API

- Data collection process:
- Start with URL:

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

- GitHub URL of the completed SpaceX API calls notebook

https://github.com/jwbrad01/testrepo



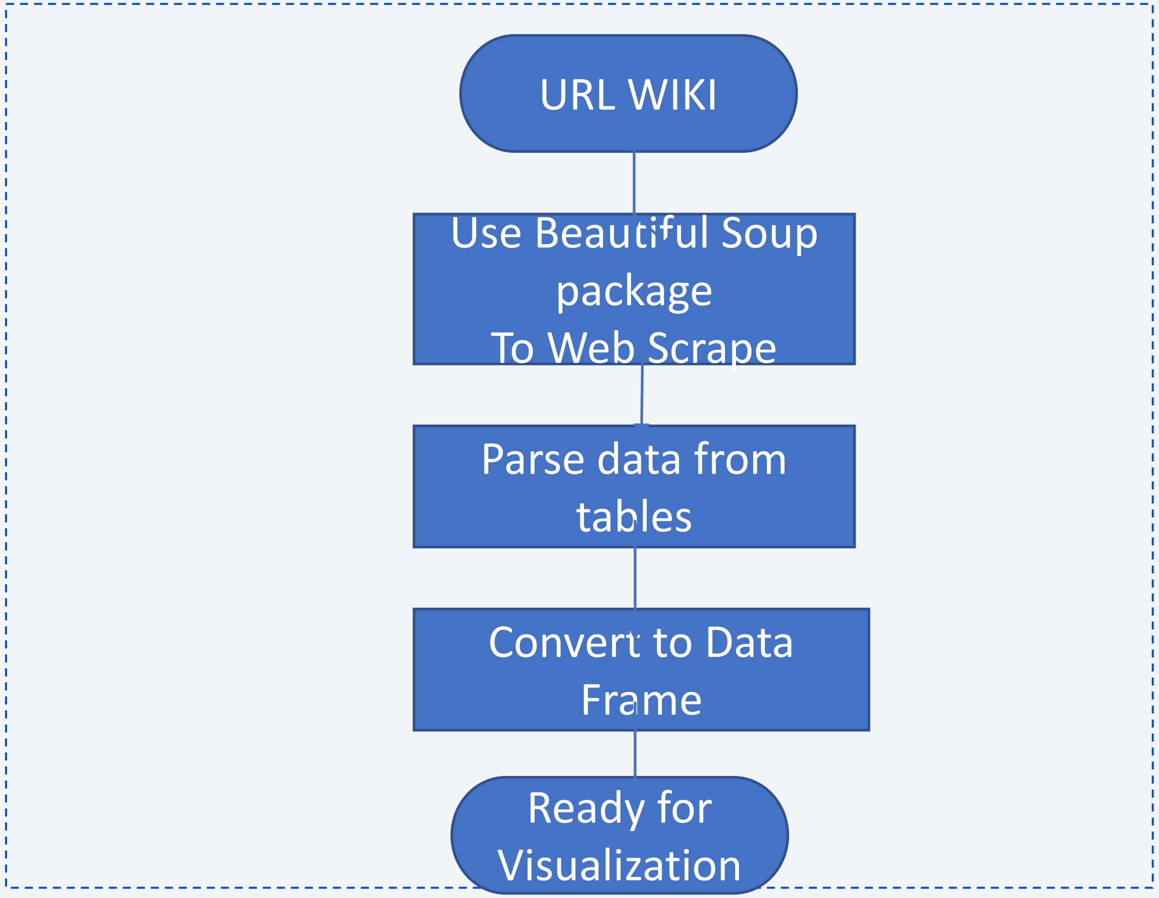
Data Collection - Scraping

- Web scraping process using and key phrases
- URL

static_url **="https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"**

- GitHub URL of the completed web scraping notebook:

https://github.com/jwbrad01/testrepo



Data Wrangling

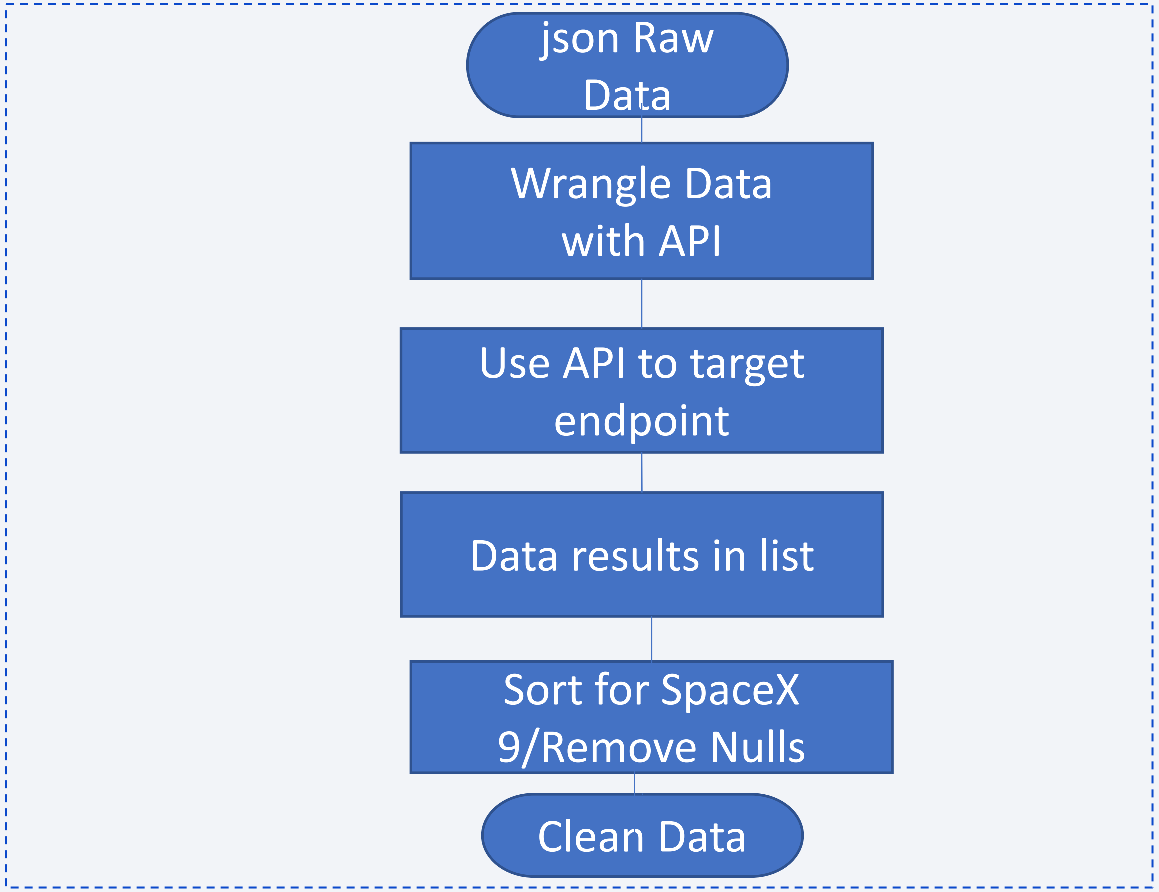
- Data Process;

  See flow chart

- Data wrangling process using key phrases *next page*

- GitHub URL of your completed data wrangling:

https://github.com/jwbrad01/testrepo



Functions to Call API

Function	Target	Endpoint
getBoosterVersion	Rocket Name	https://api.spacexdata.com/v4/rockets/
getLaunchSite	Launch Site Latitude and Longitude	https://api.spacexdata.com/v4/launchpads/
getPayLoadData	Pay Load Data Mass of Payload and Orbit	https://api.spacexdata.com/v4/payloads/
getCoresData	Cores Data Outcome of Landing	https://api.spacexdata.com/v4/cores/

EDA with Data Visualization

- Summarize what charts were plotted and why these were used those charts

  Flight number vs Payload mass – to determine success of landing

  (increase in flight number likely hood in landing success

  Payload vs Site – determines mass and relationship to location

  Success of each orbit (success and orbit type)

The purpose of this visualization tell the correlation or relationship with mass, flight and location

- GitHub URL of your completed EDA with data visualization:

  https://github.com/jwbrad01/testrepo

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed

  Listing of all launches – sorting to find the unique launches (the name)

  Identify KSC Sites

  Sum and average of payload mass

  Successful and failed launch attempts and rank them

- GitHub URL of your completed EDA with SQL:

  https://github.com/jwbrad01/testrepo

Build an Interactive Map with Folium

- Summary of what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Objects added

Mouse Position – to mouse over any point (mouse over) to get coordinates on the map and allowing drill into railway and more. By using mouse position, the Lat/Long can be used to do distance calculations

Marker clusters – has longitude and latitude brings all markings together – identify class 1 success and class 0 fail and can also identify easily which ones have failed given their similar site Lat/Long

Circles- highlight circles on coordinate for a site

Lines (poly) – allows to select one point to via a line to another point

- GitHub URL of your completed Interactive Map with Folium:

    https://github.com/jwbrad01/testrepo

Build a Dashboard with Plotly Dash

- Dashboard display with Plotly Dash – Looks at each site
- Pie Charts used to depict successful launches vs failures for the four sites
- Scatter Plots depict Outcome, Payload Mass for booster versions

GitHub URL of completed Plotly Dash lab, as an external reference and peer-review purpose (see page 40-42)

https://github.com/jwbrad01/testrepo

Predictive Analysis (Classification)

- Summary of how model was built, evaluated, improved, and found the best performing classification model – (Loaded data, built a data frame from a matrix, normalizes with preprocessing, split data (test and train), run 4 models and conducted score for accuracy

- Present model development process using key phrases and flowchart

Load and Normalize data → Split Data → Id Models and Run Models → Find Accuracy

- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

https://github.com/jwbrad01/testrepo

Results

- Exploratory data analysis results – Page 19-24
- Interactive analytics demo in screenshots Page 40-42
- Predictive analysis results Page 45

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

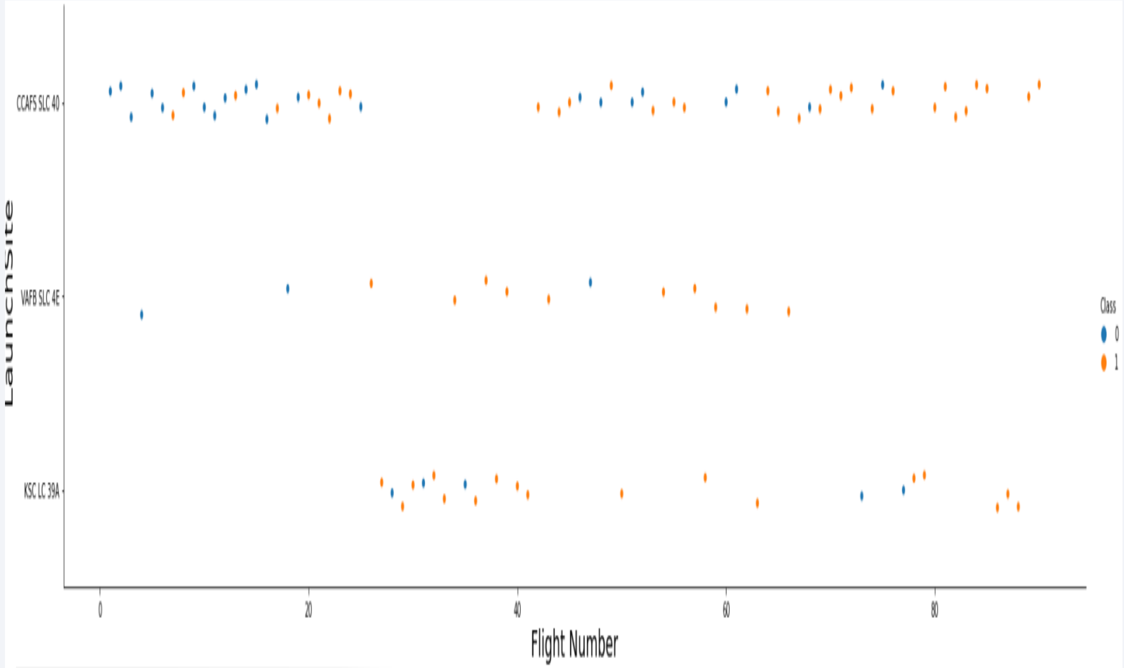
- Scatter plot of Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations:

KSCLLC39A – no launches in early flights and a good number of successful launches in mid to end from 25 to 90 flight number

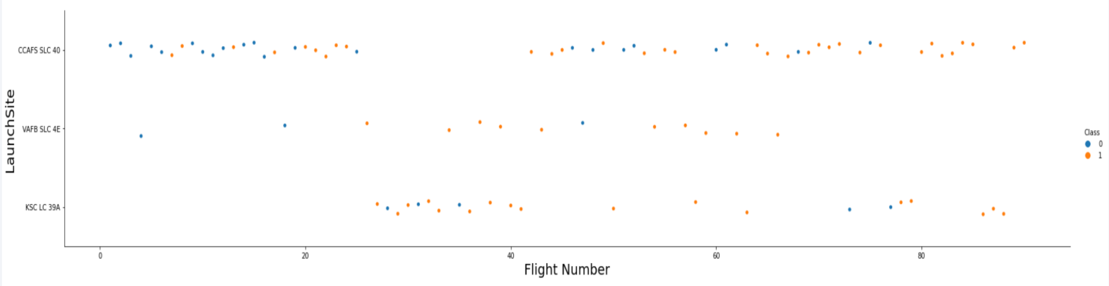
VAFB SLC 4E – one launch throughout and has few that are successful from 30 to 70 flight number

CCAFS SLC 40 – many clusters of launches in early flights and mid to late flights. Most successful launches

More launches better successful chances




```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



Payload vs. Launch Site

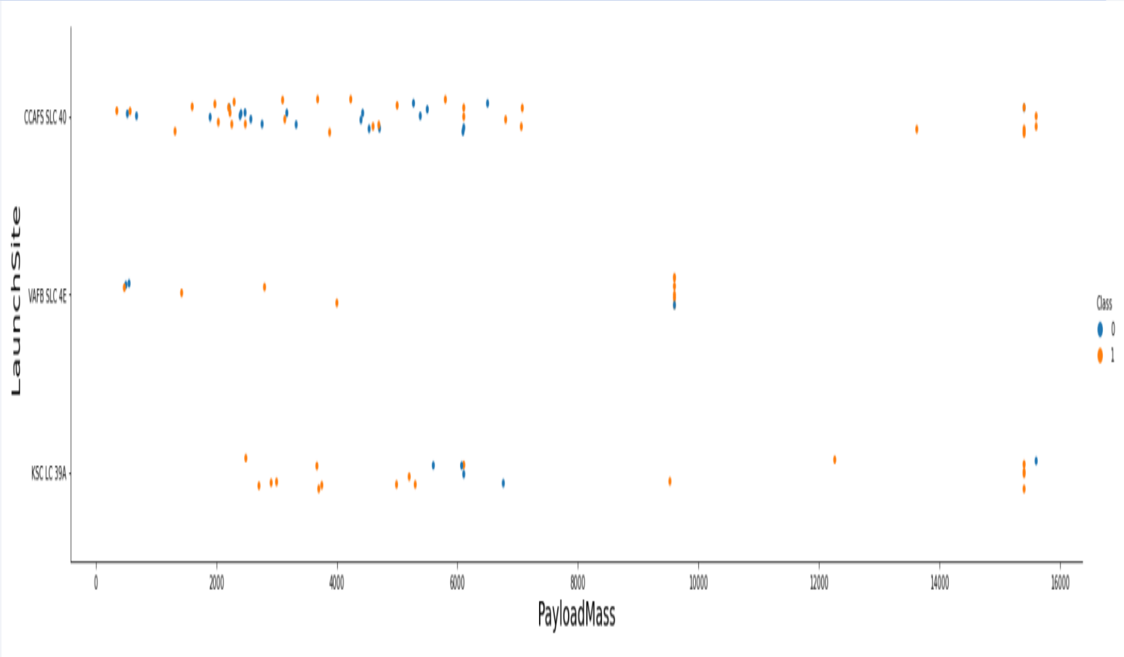
- Scatter plot of Payload vs. Launch Site

- Screenshot of the scatter plot
  KSCLLC39A – most launches occur at mass 2,000 to 6,000 and a view at 15,000

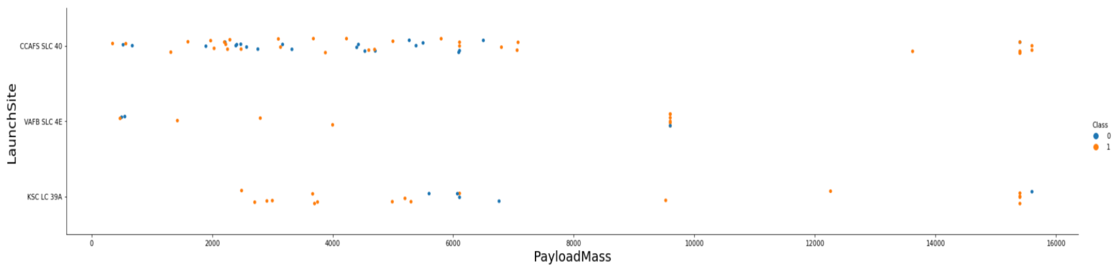
  VAFB SLC 4E – no launches for heavy load mass >10000

  CCAFS SLC 40 – launches are constant a mass up to 6000

Heavy weight could have an influence over launches

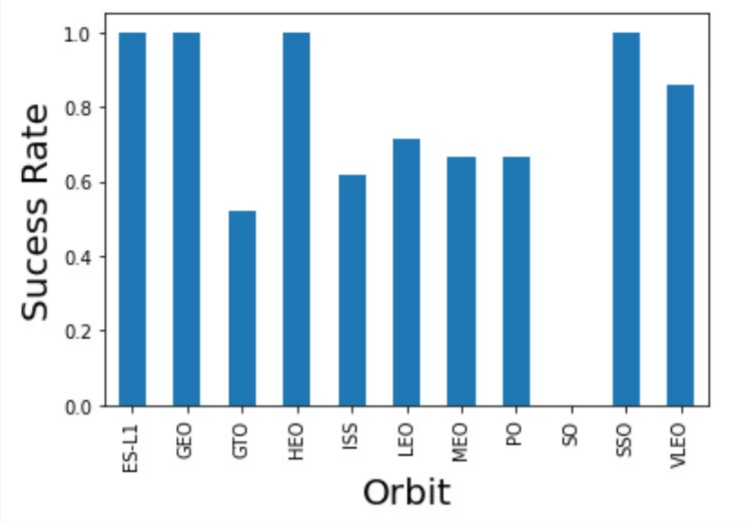



```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

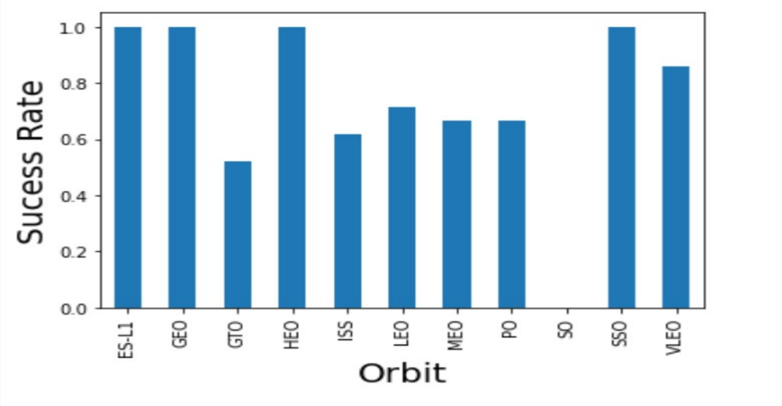


Success Rate vs. Orbit Type

- Bar chart for the success rate of each orbit type
- Screenshot of the scatter plot explanations: This is a bar chart and shows that the highest success rate is SSO, HEO, GEO, ESL1 and VLEO

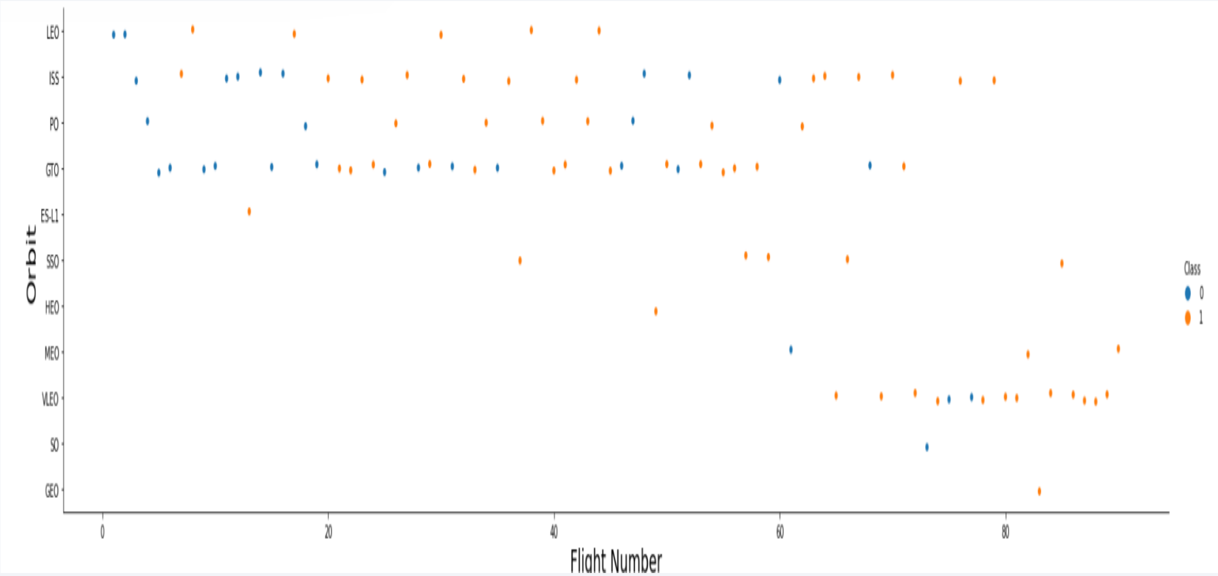


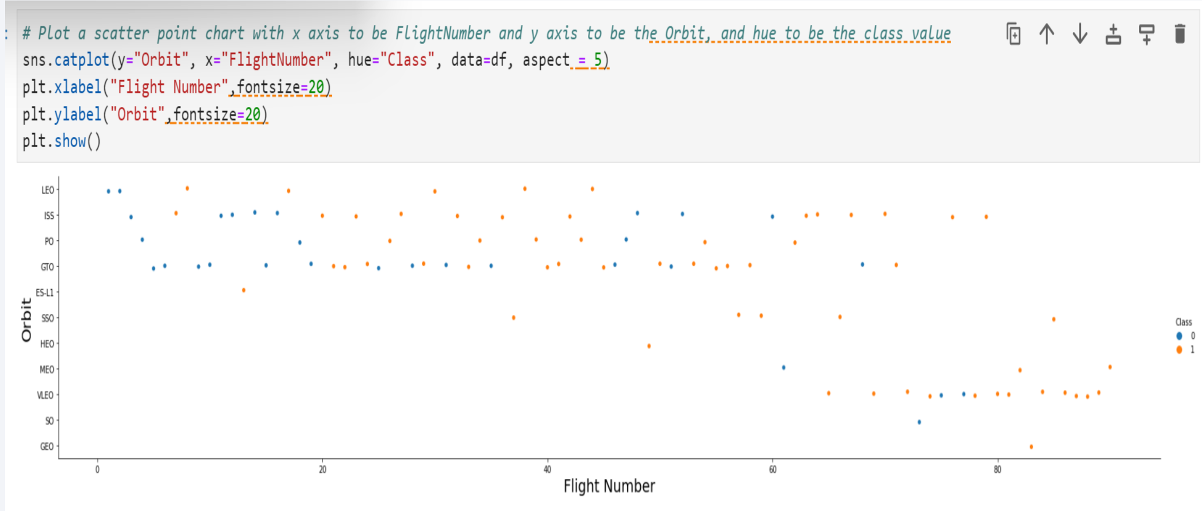

```
df.groupby(['Orbit']).mean()['Class'].plot(kind='bar')
plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Sucess Rate",fontsize=20)
plt.show()
```



Flight Number vs. Orbit Type

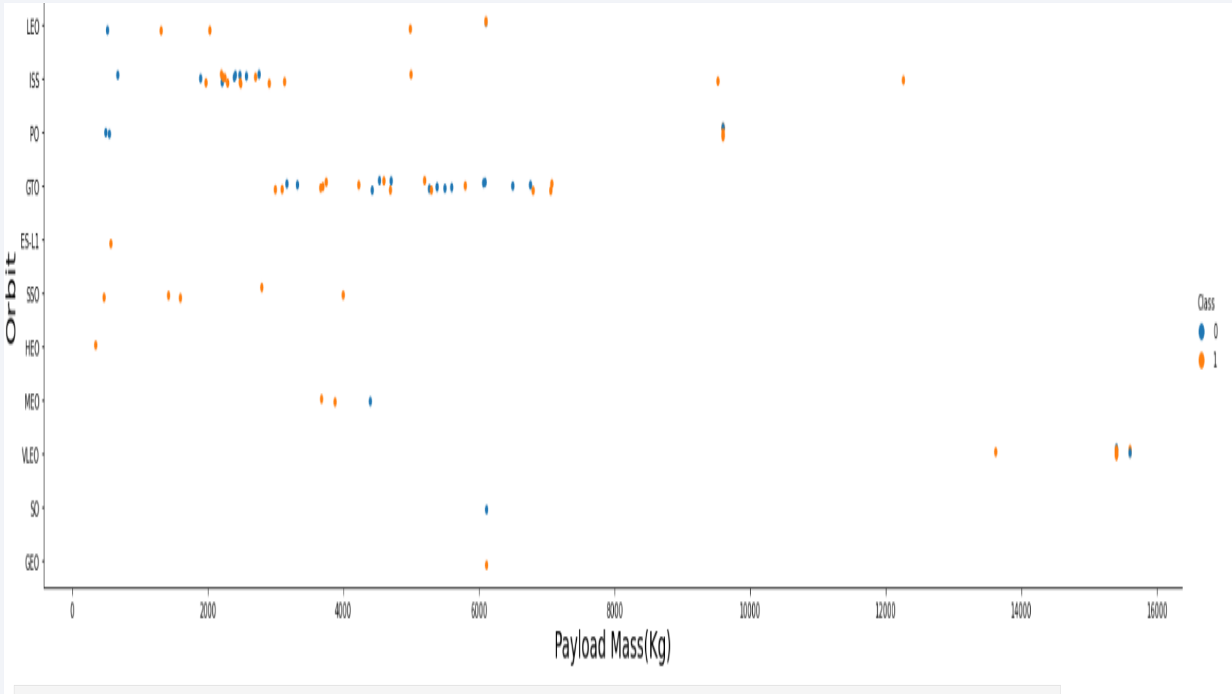
- Scatter point of Flight number vs. Orbit type
- Screenshot of the scatter plot with explanations: LEO and ISS seem to have more success in launches perhaps due to the number of flights and also SO has a good number of successful flights to further solidify this observation (>60)



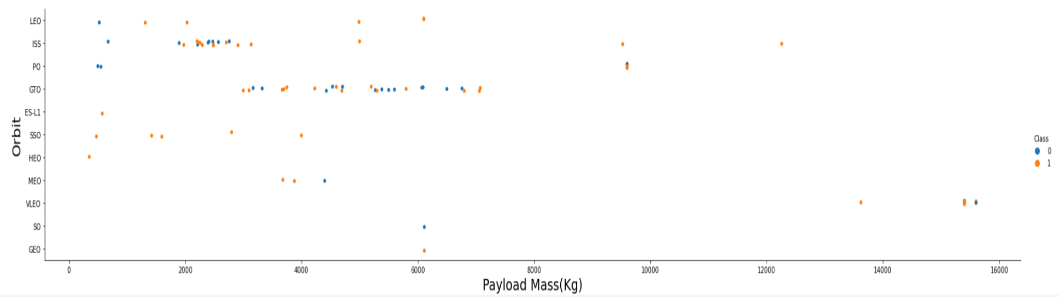


Payload vs. Orbit Type

- Scatter point of payload vs. orbit type
- Screenshot of the scatter plot with explanations: increase in payload mass there is a successful launching for LEO and ISS and for GTO there are mixed results in launching success, so it does not appear orbit is based on payload mass

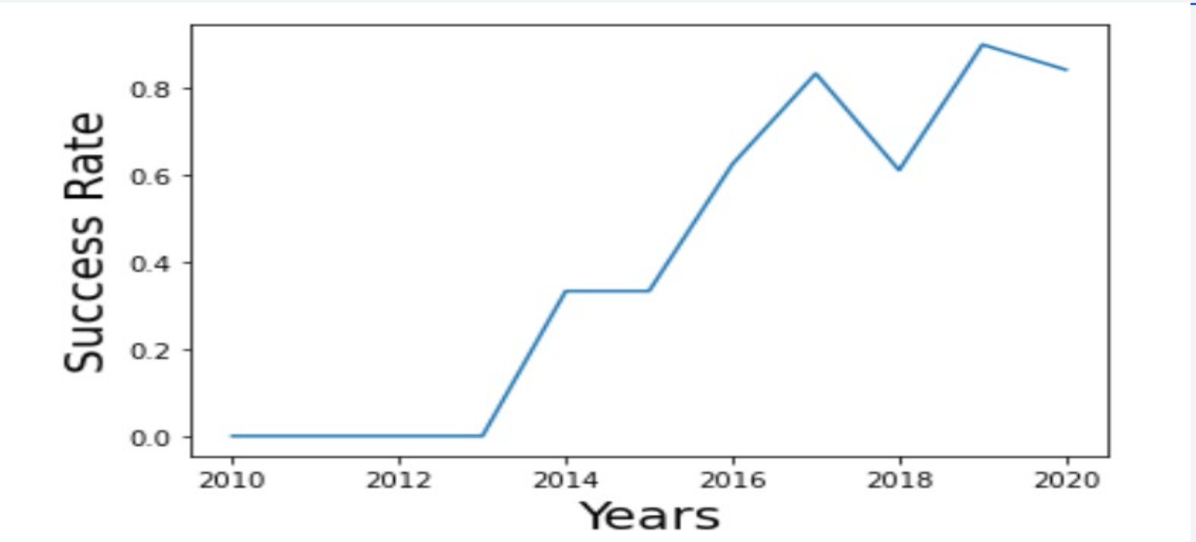



```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass(Kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



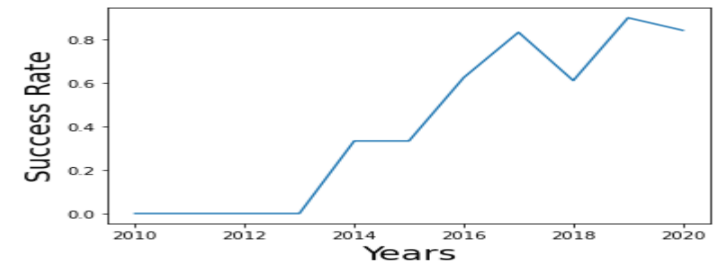
Launch Success Yearly Trend

- Line chart of yearly average success rate
- Screenshot of the line plot with explanations: The success rate continues to increase from 2013 to 2020 though there are some periods decline along the way. Perhaps the increase is due to increase launches the better the more success because of experience




```
# A function to Extract years from the date
year=[]
def Extract_year(date):
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
```

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
df['Year'] = pd.DataFrame(Extract_year(df['Date'])).astype('int')
sns.lineplot(x = df['Year'].unique() , y = df.groupby(['Year'])['Class'].mean())
plt.xlabel("Years",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



All Launch Site Names

- Names of the unique launch sites

CCAFS LC-40

CCAFS SLC-40

KSC LC-39A

VAFB SLC-4E

- Query result with a short explanation here

This query removes duplicates

```
%sql select Unique(LAUNCH_SITE) from SPACEXTABLE;
 * ibm_db_sa://lbq18727:***@125f9f61-9715-46f9-939
   sqlite:///my_data1.db
Done.
```

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

Launch Site Names Begin with 'CCA'

- 5 records listed where launch sites' names start with 'CCA'

- Query result with a short explanation here –These are the Launch sites that begin with 'CCA' and it list five as requested with the user of Like and Limit function

```
%sql SELECT LAUNCH_SITE from SPACEXTABLE where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://lbq18727:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqd
   sqlite:///my_data1.db
Done.
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Total payload carried by boosters from NASA
- Query result with a short explanation

This is the total of the weight of the payload in the table. This function added all the weights together

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTABLE;
```

```
 * ibm_db_sa://lbq18727:***@125f9f61-9715-46f9-9399-c8177b21803b.c1o
   sqlite:///my_data1.db
Done.
```

payloadmass
619967

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Query result with a short explanation here: This is the average mass weight of the payload using select avg function

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTABLE;
```

```
 * ibm_db_sa://lbq18727:***@125f9f61-9715-46f9-9399-c8177b21803b.c1
   sqlite:///my_data1.db
Done.
```

payloadmass
6138

First Successful Ground Landing Date

- Dates of the first successful landing outcome on ground pad
- Query result with a short explanation here: Did a query for the minimum successful landing using select min function

```
%sql select min(DATE) from SPACEXTABLE;
 * ibm_db_sa://lbq18727:***@125f9f61-9715
   sqlite:///my_data1.db
Done.
```

1
2010-04-06

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Query result with a short explanation here: Booster version (Booster) from Space Table and where locates the successful outcome between 4000 and 6000

```
%sql select BOOSTER_VERSION from SPACEXTABLE where LANDING__OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
```

```
 * ibm_db_sa://lbq18727:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
   sqlite:///my_data1.db
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes
- Query result with a short explanation here: This is a list of a count of 1) Mission Outcomes, 2)Success and 3) Failure Outcomes in that order

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTABLE GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://lbq18727:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.ap
   sqlite:///my_data1.db
Done.
```

missionoutcomes
1
99
1

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass
- Query result with a short explanation here: This is a list of booster versions with highest in weight for payload

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTABLE where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE);
```

```
 * ibm_db_sa://lbq18727:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludb
   sqlite:///my_data1.db
Done.
```

boosterversion
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

- Displayed is the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- Query result with a short explanation here: This code gives the 2017 launch records by month names of successful records

```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTABLE where EXTRACT(YEAR FROM DATE)='2017';
```

```
 * ibm_db_sa://lbq18727:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30426/bludl
   sqlite:///my_data1.db
```

1	mission_outcome	booster_version	launch_site
1	Success	F9 FT B1029.1	VAFB SLC-4E
2	Success	F9 FT B1031.1	KSC LC-39A
3	Success	F9 FT B1030	KSC LC-39A
3	Success	F9 FT B1021.2	KSC LC-39A
1	Success	F9 FT B1032.1	KSC LC-39A
5	Success	F9 FT B1034	KSC LC-39A
3	Success	F9 FT B1035.1	KSC LC-39A
6	Success	F9 FT B1029.2	KSC LC-39A
6	Success	F9 FT B1036.1	VAFB SLC-4E
5	Success	F9 FT B1037	KSC LC-39A
8	Success	F9 B4 B1039.1	KSC LC-39A
8	Success	F9 FT B1038.1	VAFB SLC-4E
7	Success	F9 B4 B1040.1	KSC LC-39A
9	Success	F9 B4 B1041.1	VAFB SLC-4E
11	Success	F9 FT B1031.2	KSC LC-39A
10	Success	F9 B4 B1042.1	KSC LC-39A
12	Success	F9 FT B1035.2	CCAFS SLC-40
12	Success	F9 FT B1036.2	VAFB SLC-4E

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

- Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order
- Query result with a short explanation: This query allows to select a column from the table with the descending date order and the outcomes

```
%sql SELECT LANDING__OUTCOME FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

```
 * ibm_db_sa://lbq18727:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3042
   sqlite:///my_data1.db
Done.
```

landing__outcome
No attempt
Success (ground pad)
Success (ground pad)
Success (drone ship)
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Controlled (ocean)

Section 3

Launch Sites Proximities Analysis

Folium Map of Launch Sites

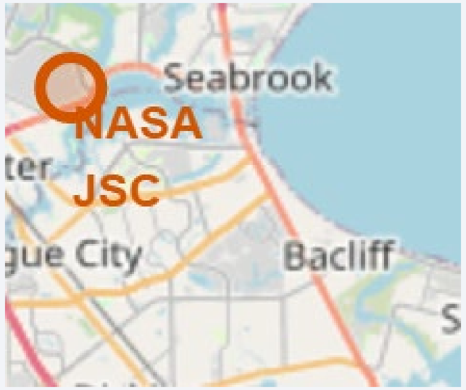
- Folium map of Launch Sites

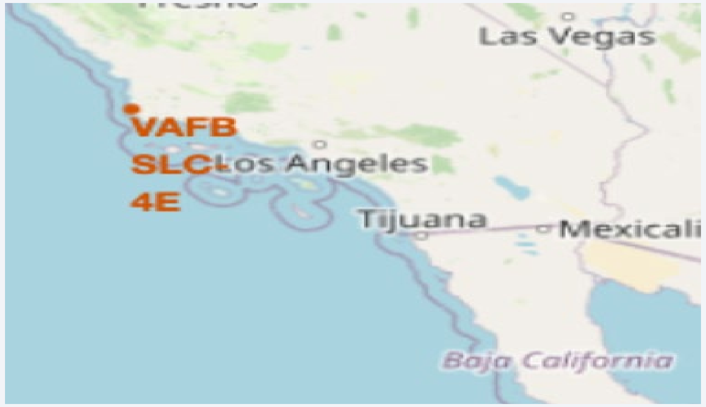
  Long/Lats

- Screenshots of site locations in both Florida and California

- Important elements and findings on the screenshot: In mapping the 4 sites based on Lat/Long by the coastline

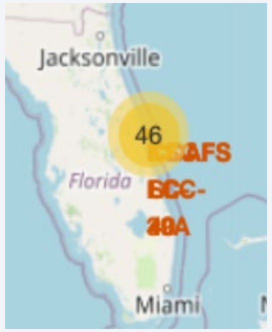
	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745



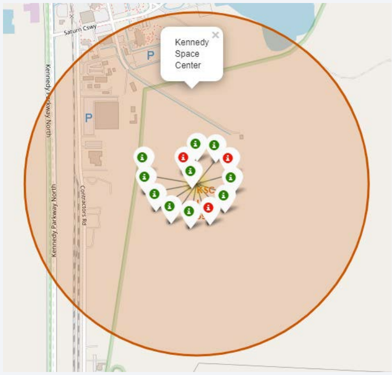


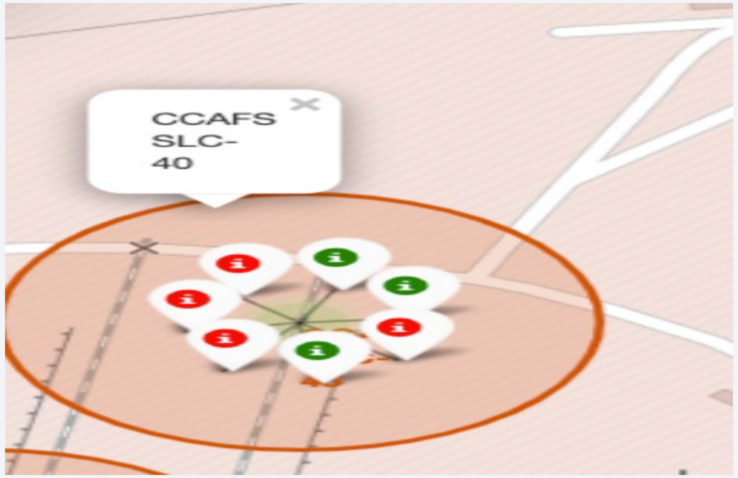
Folium Map of Launch Outcomes

- Folium Map Launch Outcomes which are identified by success Class 1 and failed Class 0
- Folium map shows color labeled outcome
- Successful Launches were KSCL LL-39A and two CCAFS SLC-40



	Launch Site	Lat	Long	class	marker_color
46	KSC LC-39A	28.573255	-80.646895	1	green
47	KSC LC-39A	28.573255	-80.646895	1	green
48	KSC LC-39A	28.573255	-80.646895	1	green
49	CCAFS SLC-40	28.563197	-80.576820	1	green
50	CCAFS SLC-40	28.563197	-80.576820	1	green
51	CCAFS SLC-40	28.563197	-80.576820	0	red
52	CCAFS SLC-40	28.563197	-80.576820	0	red
53	CCAFS SLC-40	28.563197	-80.576820	0	red
54	CCAFS SLC-40	28.563197	-80.576820	1	green
55	CCAFS SLC-40	28.563197	-80.576820	0	red

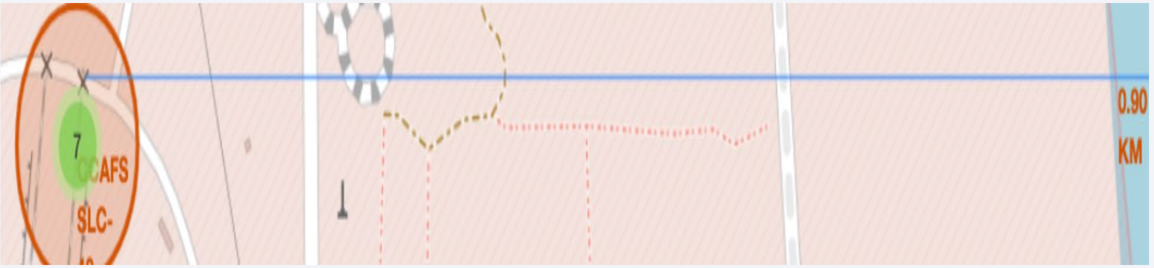




Folium Map of Proximity and Distance

- Folium Map Proximity and Distance
- Folium map and showing the selected launch site to its proximities of railway, highway, coastline, and distance calculated and displayed
- Proximity of the site to coastline

```
# approximate radius of earth in km
R = 6373.0

lat1 = radians(lat1)
lon1 = radians(lon1)
lat2 = radians(lat2)
lon2 = radians(lon2)

dlon = lon2 - lon1
dlat = lat2 - lat1

a = sin(dlat / 2)**2 + cos(lat1) * cos(lat2) * sin(dlon / 2)**2
c = 2 * atan2(sqrt(a), sqrt(1 - a))

distance = R * c
```



Section 4

Build a Dashboard with Plotly Dash

Total Successful Launch Site for Space X

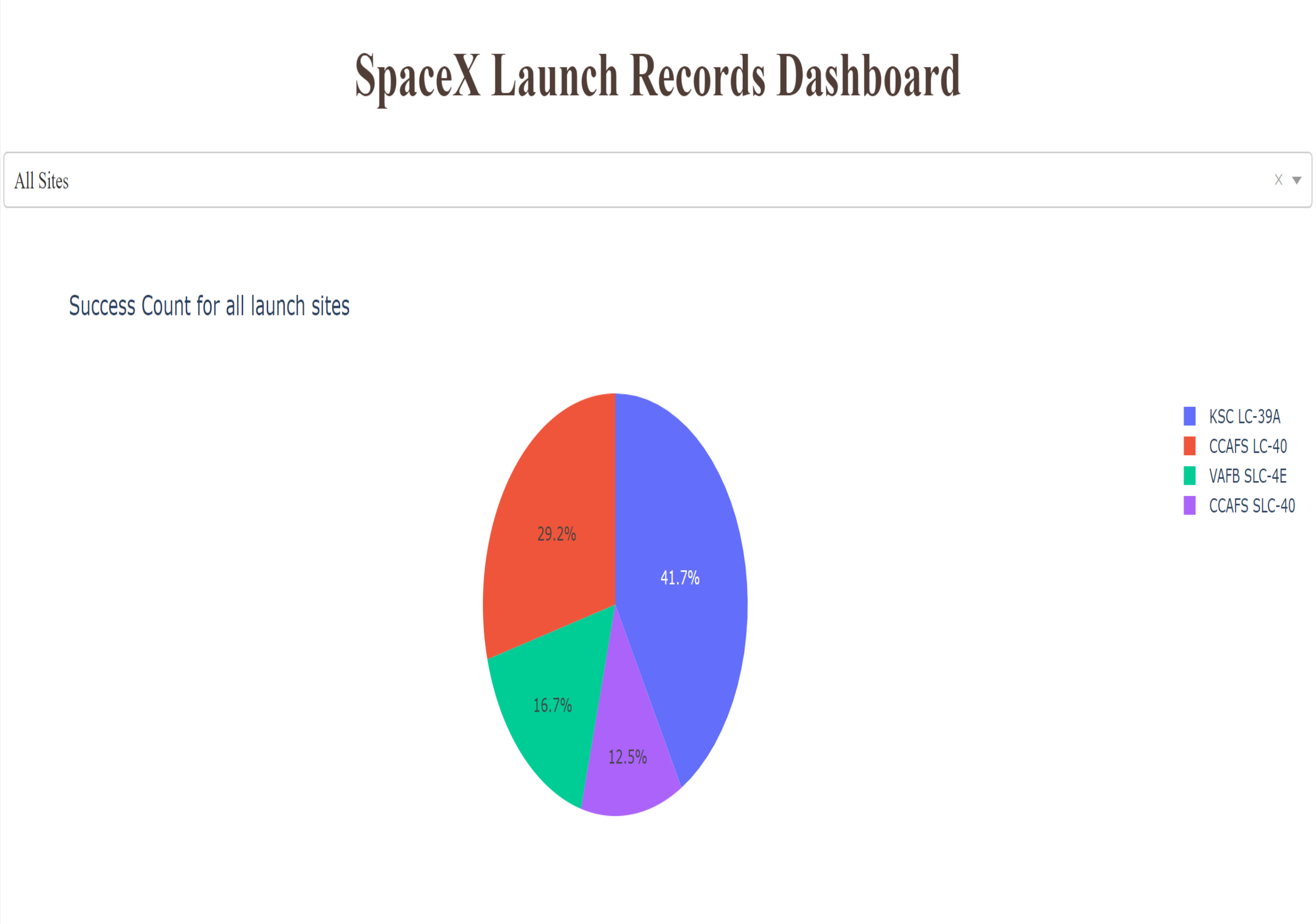
- Total Successful Launch Site for Space X

  All Sites

- Screenshot of launch success count for all sites, in a pie chart

- Important elements and findings on the screenshot

KSC LLC – 39A has the highest number of successes counts at 41.7%

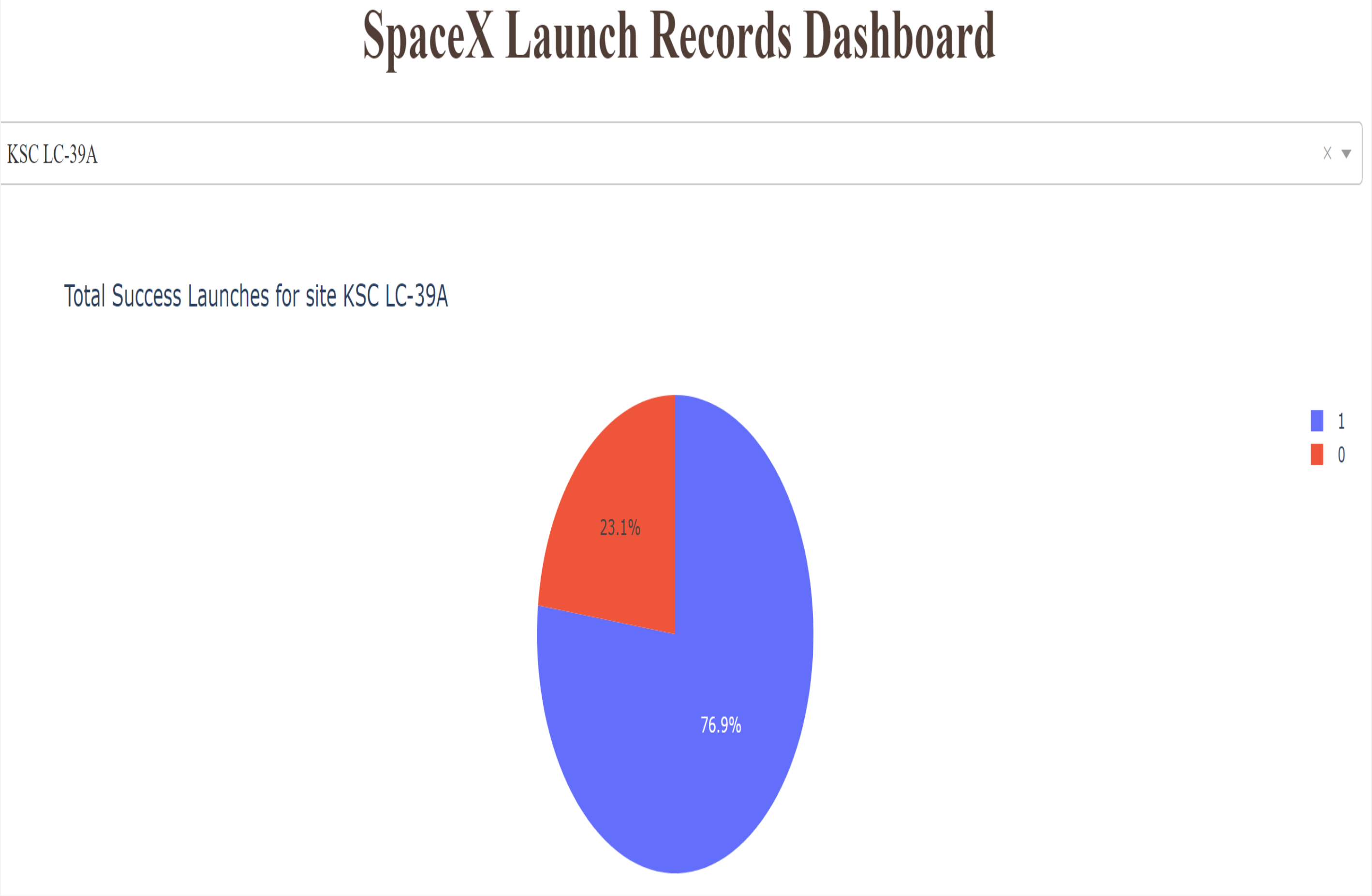


Launch Site With Highest Launches

- Highest Launches

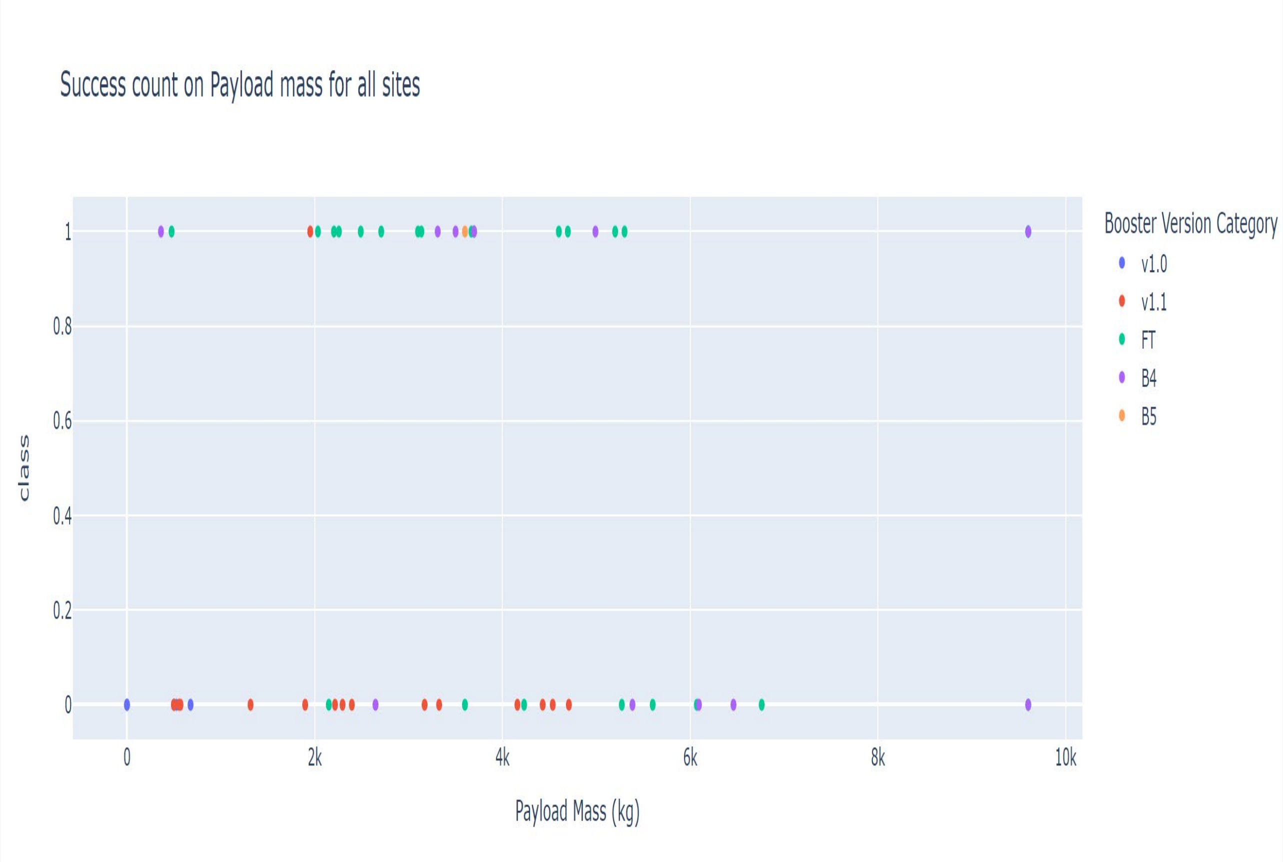
- Screenshot of the pie chart for the launch site with highest launch success ratio

- Important elements and findings on the chart : This is the launch site with the highest success ratio KSC LLC -39A 10:3 (1 = success- blue and 0 = failure – red)



Launch Outcomes

- Launch Outcomes
- Screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Important elements and findings on the screenshot, B4 has the most success with based on weight which was up to 9K. FT is next with a range of 0-7K with successes



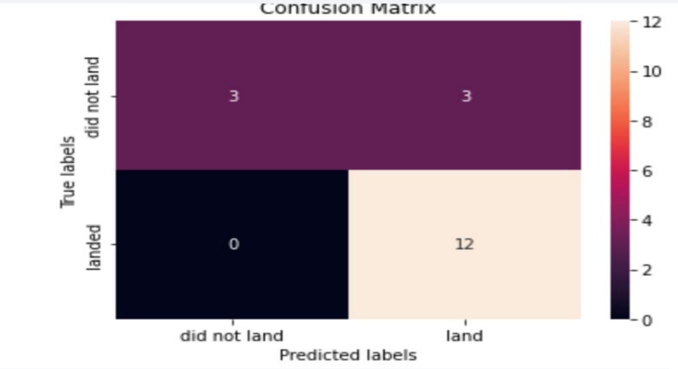
Section 5

Predictive Analysis (Classification)

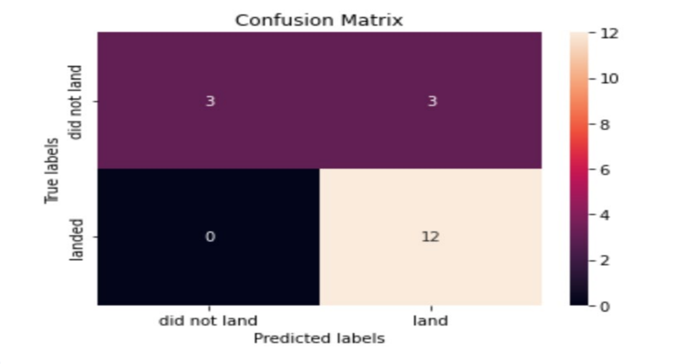
Classification Accuracy

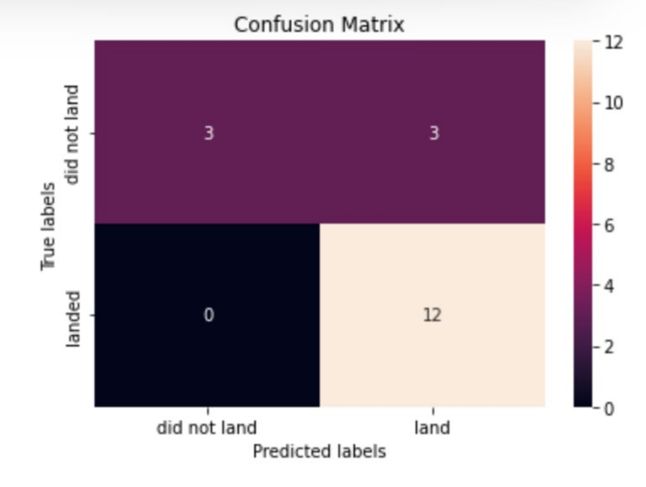
- Visualize the built model accuracy for all built classification models, in a bar chart
- Logistic regression
- SVM
- Decision Tree
- KNN
- Find which model has the highest classification accuracy

Logistic Regression

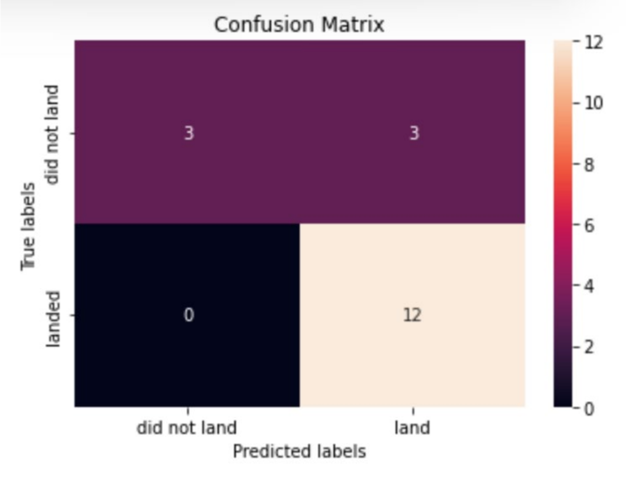


SVM





Decision Tree



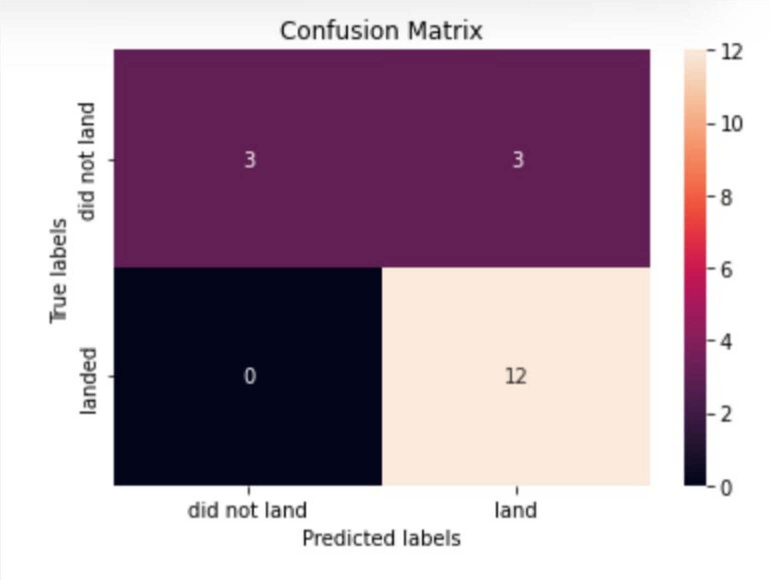
KNN

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation
- The Decision Tree is

the most accurate

*.0888888888888




```python
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()

grid_search = GridSearchCV(tree, parameters, cv=10)
tree_cv = grid_search.fit(X_train, Y_train)
```

```python
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_dept
10, 'splitter': 'random'}
accuracy : 0.8888888888888888
```

* Note results may vary because analysis was performed in Skills Network Labs and not Watson IBM Cloud

Conclusions

- Visualization based on Correlation and Relationship – Payload vs Orbit type success seem to be based on mass, increase in launch successes increases from 2013 to 2020 and increase in trend could be due to the gain of experience over time and five orbit types had the highest success rate
- Data Querying pointed out information in the data for successful launches in the data for booster versions and launch sites and some basic statistics (the sum and average mass and min successful landing)
- Data Visualizations shows that KSC LLC -39A site had the most successful launches
- Model Classification depicts that the Decision Tree has the best performance and accurate model that can be used to determine the first stage successful landing

Appendix

- Relevant assets:
- Python code snippets – Filter dataframe and **Web Scraping**

Task 2: Filter the dataframe to only include `Falcon 9` launches

Finally we will remove the Falcon 1 launches keeping only the Falcon 9 launches. Filter the data dataframe using the `BoosterVersion` column to only keep the Falcon 9 launches. Save the filtered data to a new dataframe called `data_falcon9`.

```
[52]: # Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data[data['BoosterVersion'] == 'Falcon 9']
data_falcon9
```

[52]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	Reuse
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a
soup = BeautifulSoup(response,"html.parser")  # create a soup
```

- SQL queries – How to load data SQL lite and Finding Unique Values

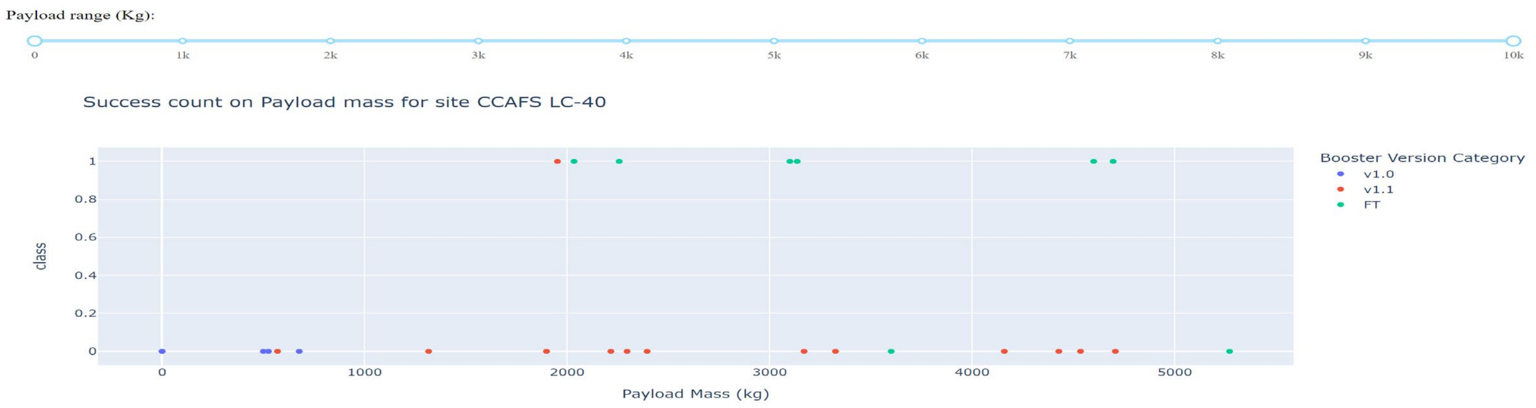
```
%sql ibm_db_sa://my-username:my-pass

[2]: %load_ext sql

[3]: # Remember the connection string is of the
# %sql ibm_db_sa://my-username:my-password@
%sql ibm_db_sa://vvn23406:m4JWx1mzgyL7ObAI@
```

```
%sql select Unique(LAUNCH_SITE) from SPACEXTABLE;
 * ibm_db_sa://lbq18727:***@125f9f61-9715-46f9-93
   sqlite:///my_data1.db
Done.
```

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

Appendix

- Chart- Lowest Payload Success Count



- Notebook outputs, or data sets created during this project

Filtered and Sorted Excel Spreadsheet Prior to DB2

	A	B	C	D	E	F	G	H	I	J	K
1	Date	Time (UTC)	Booster_Ve	Launch_Sit	Payload	PAYLOAD_I	Orbit	Customer	Mission_Ou	Landing _Outcome	
2	6/4/2010	18:45:00	F9 v1.0 B0	CCAFS LC-4	Dragon Spa	0	LEO	SpaceX	Success	Failure (parachute)	
3	12/8/2010	15:43:00	F9 v1.0 B0	CCAFS LC-4	Dragon der	0	LEO (ISS)	NASA (COT	Success	Failure (parachute)	
4	22/05/201	7:44:00	F9 v1.0 B0	CCAFS LC-4	Dragon der	525	LEO (ISS)	NASA (COT	Success	No attempt	
5	10/8/2012	0:35:00	F9 v1.0 B0	CCAFS LC-4	SpaceX CRS	500	LEO (ISS)	NASA (CRS)	Success	No attempt	
6	3/1/2013	15:10:00	F9 v1.0 B0	CCAFS LC-4	SpaceX CRS	677	LEO (ISS)	NASA (CRS)	Success	No attempt	
7	29/09/201	16:00:00	F9 v1.1 B1	VAFB SLC-4	CASSIOPE	500	Polar LEO	MDA	Success	Uncontrolled (ocean)	
8	12/3/2013	22:41:00	F9 v1.1	CCAFS LC-4	SES-8	3170	GTO	SES	Success	No attempt	

Machine Learning Model Accuracy for:
KNN, Logistic, SMV

accuracy : 0.8472222222222222

And the test data accuracy:

0.8333333333333334

Thank you!