Data Science Capstone project

<Ali Hassan Mirza>

<August 30, 2021>

Outline

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

Executive Summary

- Summary of methodologies

Collected data from public SpaceX API and SpaceX Wikipedia page. Created labels column 'class' which classifies successful landings. Explored data using SQL, visualization, folium maps, and dashboards. Gathered relevant columns to be used as features. Changed all categorical variables to binary using one hot encoding. Standardized data and used GridSearchCV to find best parameters for machine learning models. Visualize accuracy score of all models

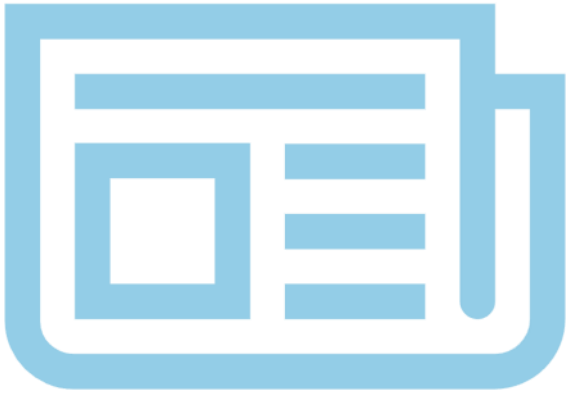
- Summary of all results

Four machine learning models were produced: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors. All produced similar results with accuracy rate of about 83.33%. All models over predicted successful landings. More data is needed for better model determination and accuracy

Introduction

- Project background and context
  - Commercial Space Age Evolution
  - Space X has best pricing as compared to competitors
  - Largely due to ability to recover part of rocket (Stage 1)
  - Space Y wants to compete with Space X
- Problems you want to find answers
  - Space Y tasks us to train a machine learning model to predict successful Stage 1 recovery

Methodology



- Data collection methodology:
  - Combined data from SpaceX public API and SpaceX Wikipedia page
- Perform data wrangling
  - Classifying true landings as successful and unsuccessful otherwise
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tuned models using GridSearchCV

Methodology

Data collection

- Data collection process involved a combination of API requests from Space X public API and web scraping data from a table in Space X's Wikipedia entry.

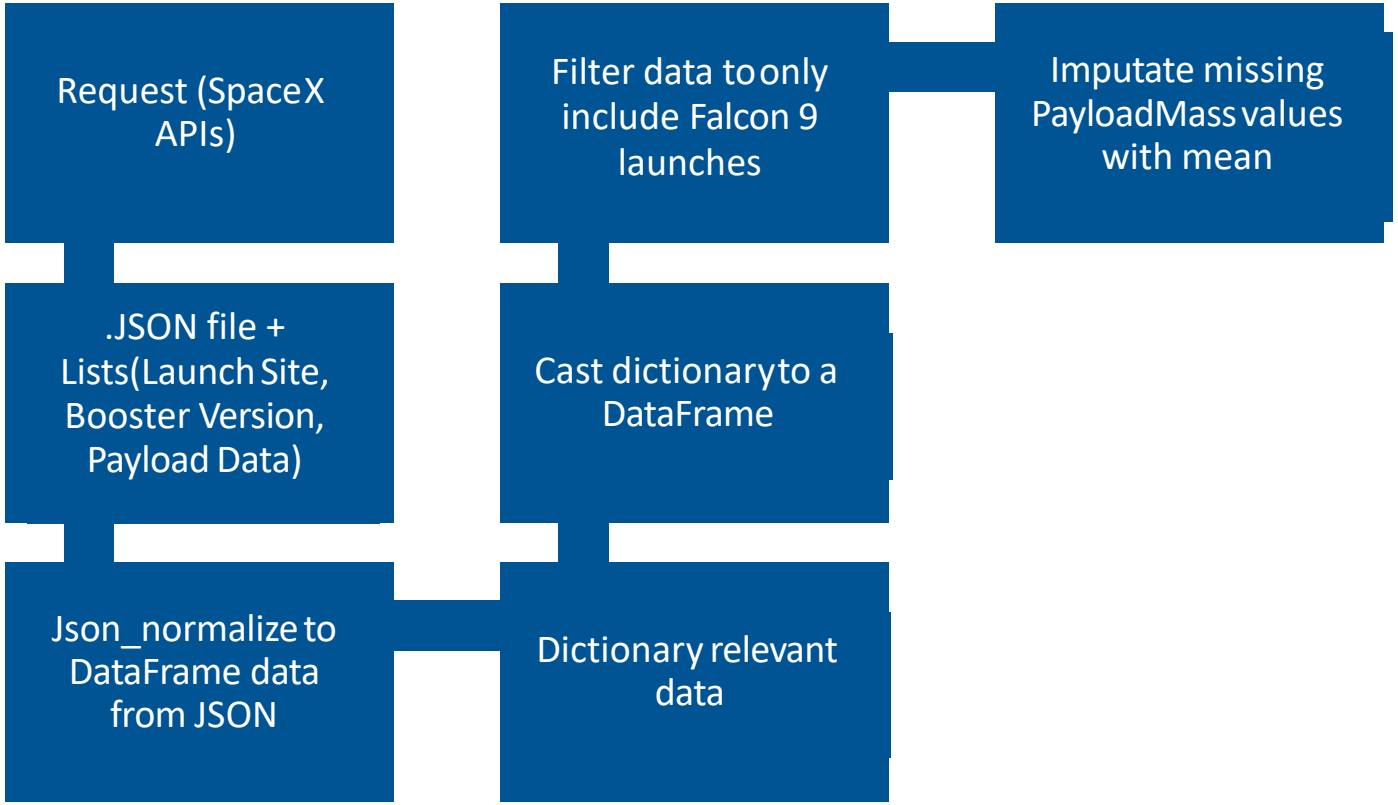
- The next slide will show the flowchart of data collection from API and the one after will show the flowchart of data collection from webscraping.

- Space X API Data Columns:
  - FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins,Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

- Wikipedia Webscrape Data Columns:
  - Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data collection – SpaceX API

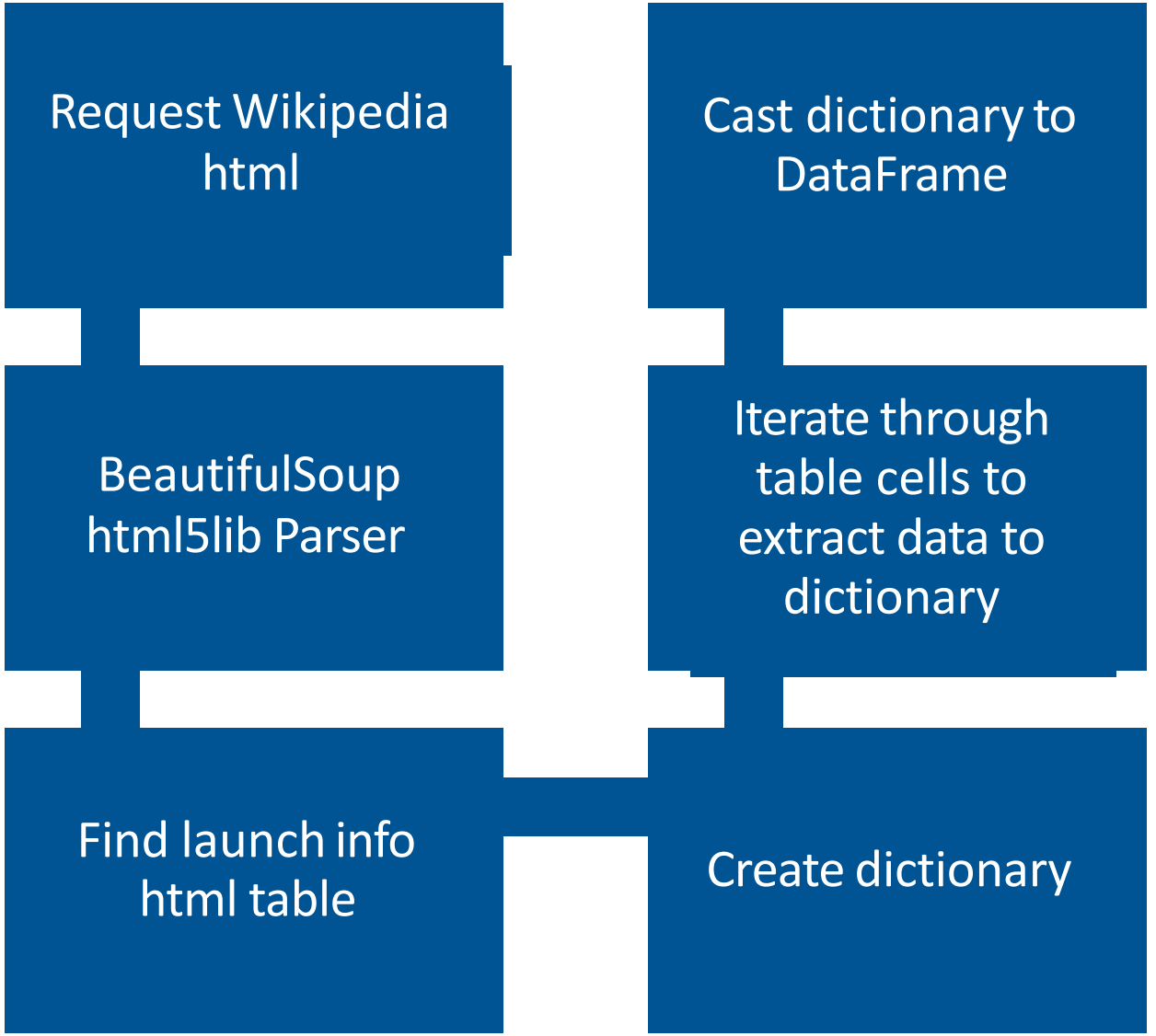
https://github.com/alimirzah/IBM_Data_Science_Certification/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data collection – Web scraping

Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

https://github.com/alimirzah/IBM_Data_Science_Certification/blob/main/jupyter-labs-webscraping.ipynb



Data wrangling

- Create a training label with landing outcomes where successful = 1 & failure = 0.
- Outcome column has two components: ‘Mission Outcome’ and ‘Landing Location’
- New training label column ‘class’ with a value of 1 if ‘Mission Outcome’ is True and 0 otherwise.
- Value Mapping:
  - True ASDS, True RTLS, & True Ocean – set to -> 1
  - None None, False ASDS, None ASDS, False Ocean, False RTLS – set to -> 0
- GitHub url:
  - https://github.com/alimirzah/IBM_Data_Science_Certification/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with data visualization

- Exploratory Data Analysis performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class and Year.
- Plots Used:
  - Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit vs. Success Rate, Flight Number vs. Orbit, Payload vs Orbit, and Success Yearly Trend
- Scatter plots, line charts, and bar plots were used to compare relationships between variables to decide if a relationship exists so that they could be used in training the machine learning model
- GitHub url:
  - https://github.com/alimirzah/IBM_Data_Science_Certification/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Loaded data set into IBM DB2 Database.
- Queried using SQL Python integration.
- Queries were made to get a better understanding of the dataset.
- Queried information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

- GitHub url:
  - https://github.com/alimirzah/IBM_Data_Science_Certification/blob/main/jupyter-labs-eda-sql-coursera.ipynb

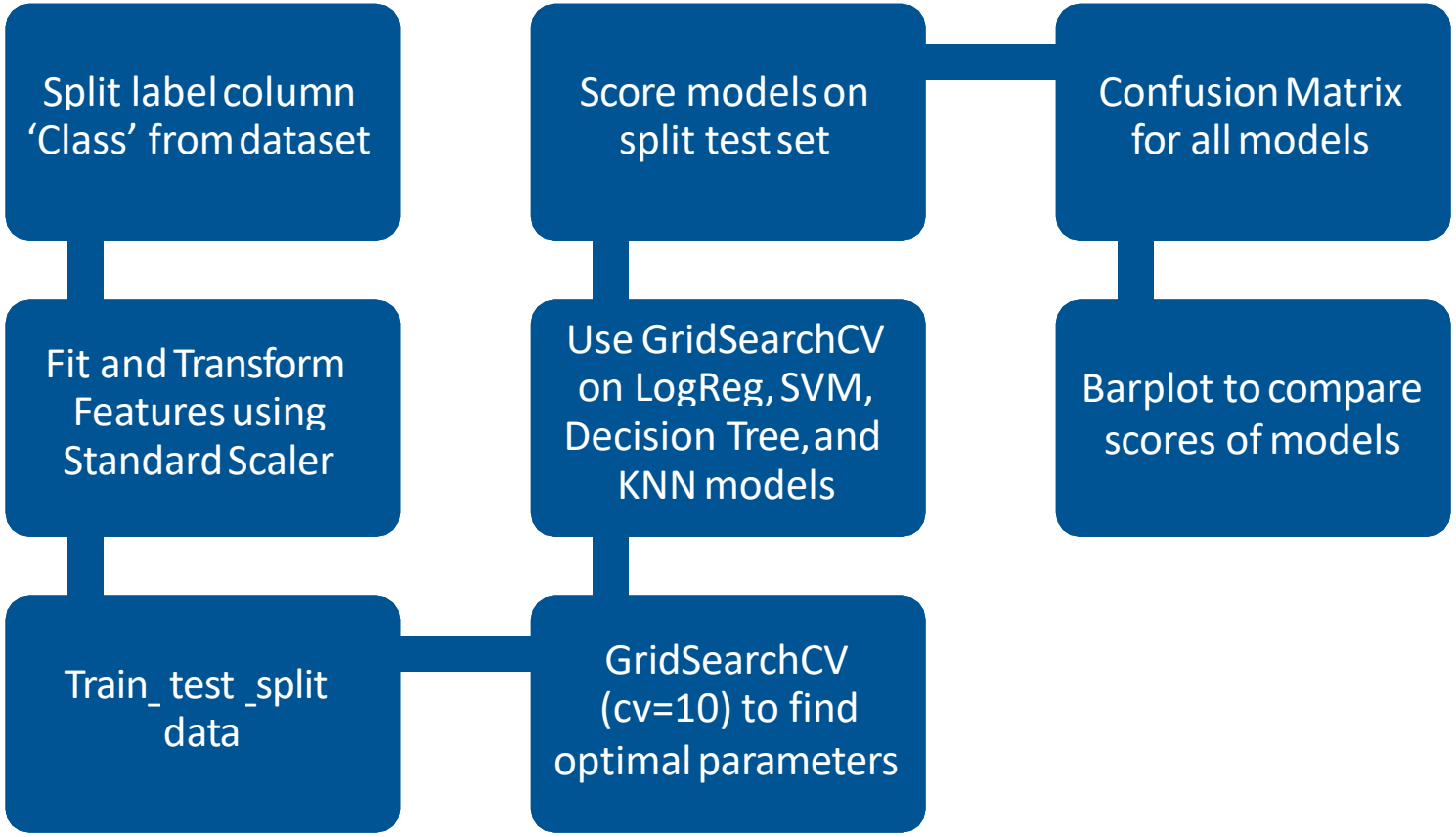
Build an interactive map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.
- This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.
- GitHub url:
  - https://github.com/alimirzah/IBM_Data_Science_Certification/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

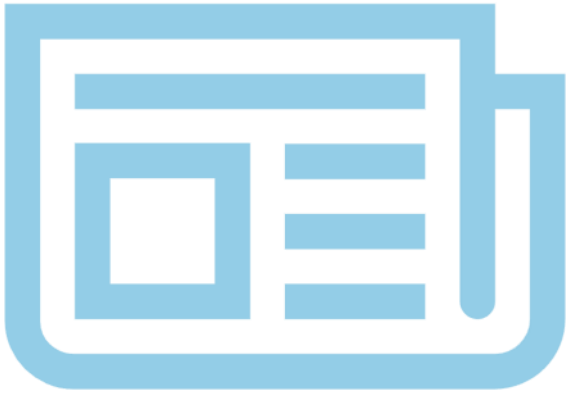
- Summarize what plots/graphs and interactions you have added to a dashboard
- Dashboard includes a pie chart and a scatter plot.
- Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.
- Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.
- The pie chart is used to visualize launch site success rate.
- The scatter plot can help us see how success varies across launch sites, payload mass, and booster version category.
- GitHub url:
  - https://github.com/alimirzah/IBM_Data_Science_Certification/blob/main/Interactive_Dashboard.py

Predictive analysis (Classification)



GitHub url: https://github.com/alimirzah/IBM_Data_Science_Certification/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results



- Exploratory data analysis results
  - In upcoming slides
- Interactive analytics demo in screenshots
  - In upcoming slides
- Predictive analysis results
  - In upcoming slides

EDA with Visualization

Flight Number vs. Launch Site

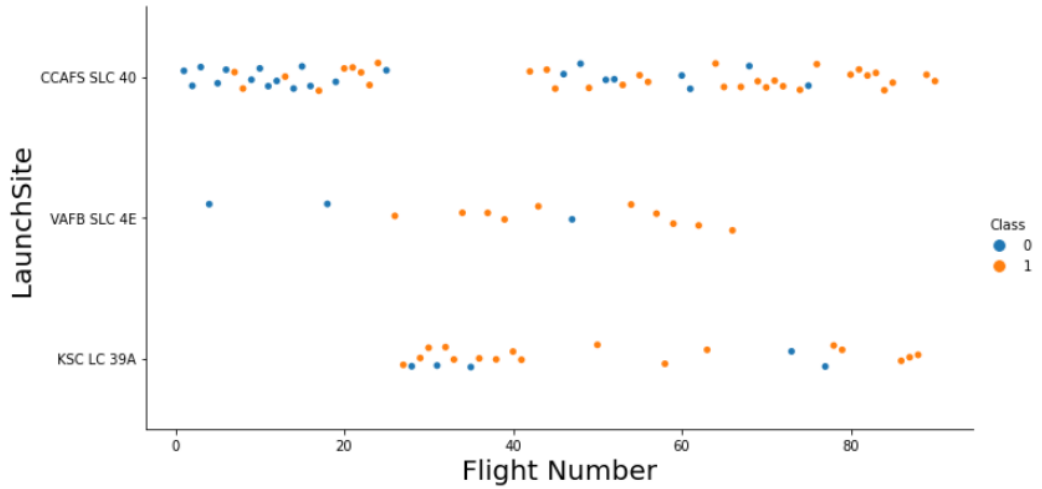
Show a scatter plot of Flight Number vs. Launch Site

Show the screenshot of the scatter plot with explanations

Orange indicates successful launch; Blue indicates unsuccessful launch.

Graphic suggests an increase in success rate over time (indicated in Flight Number). Likely a big breakthrough around flight 20 which significantly increased success rate. CCAFS appears to be the main launch site as it has the most volume

```
In [12]: # 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 = 2)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



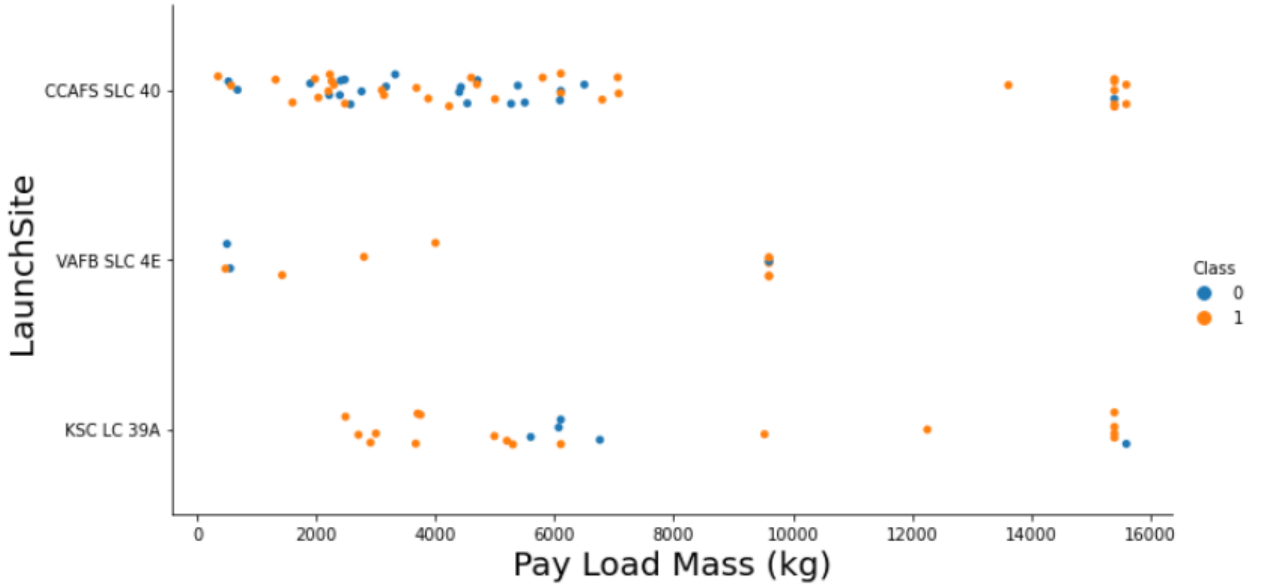
Payload vs. Launch Site

Show a scatter plot of Payload vs. Launch Site

Show the screenshot of the scatter plot with explanations

Payload mass appears to fall mostly between 0-6000 kg. Different launch sites also seem to use different payload mass.

```
In [14]: # Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, a
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 2)
plt.xlabel("Pay Load Mass (kg)",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



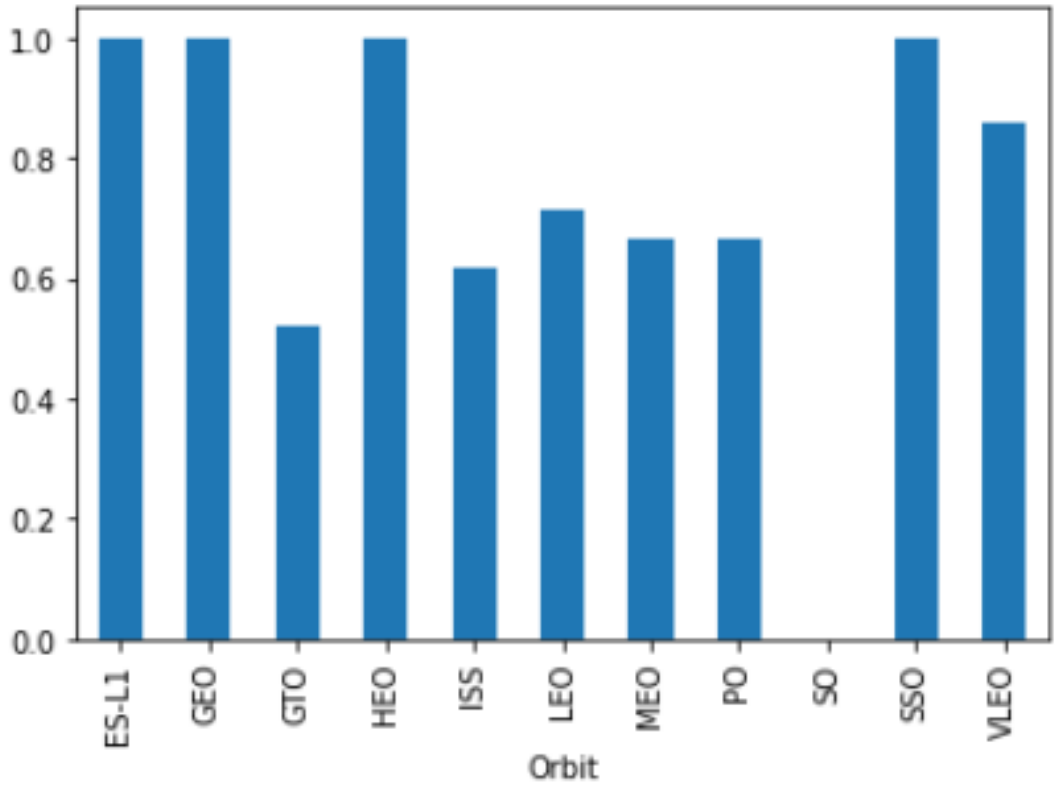
Success rate vs. Orbit type

Show a barchart for the success rate of each orbit type

Show the screenshot of the scatter plot with explanations

```
[4]: df.groupby("Orbit").mean()['Class'].plot(kind='bar')

[4]: <matplotlib.axes._subplots.AxesSubplot at 0x1ecf365ab
```



ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis)

SSO (5) has 100% success rate

VLEO (14) has decent success rate and attempts

SO (1) has 0% success rate

GTO (27) has the around 50% success rate but largest sample

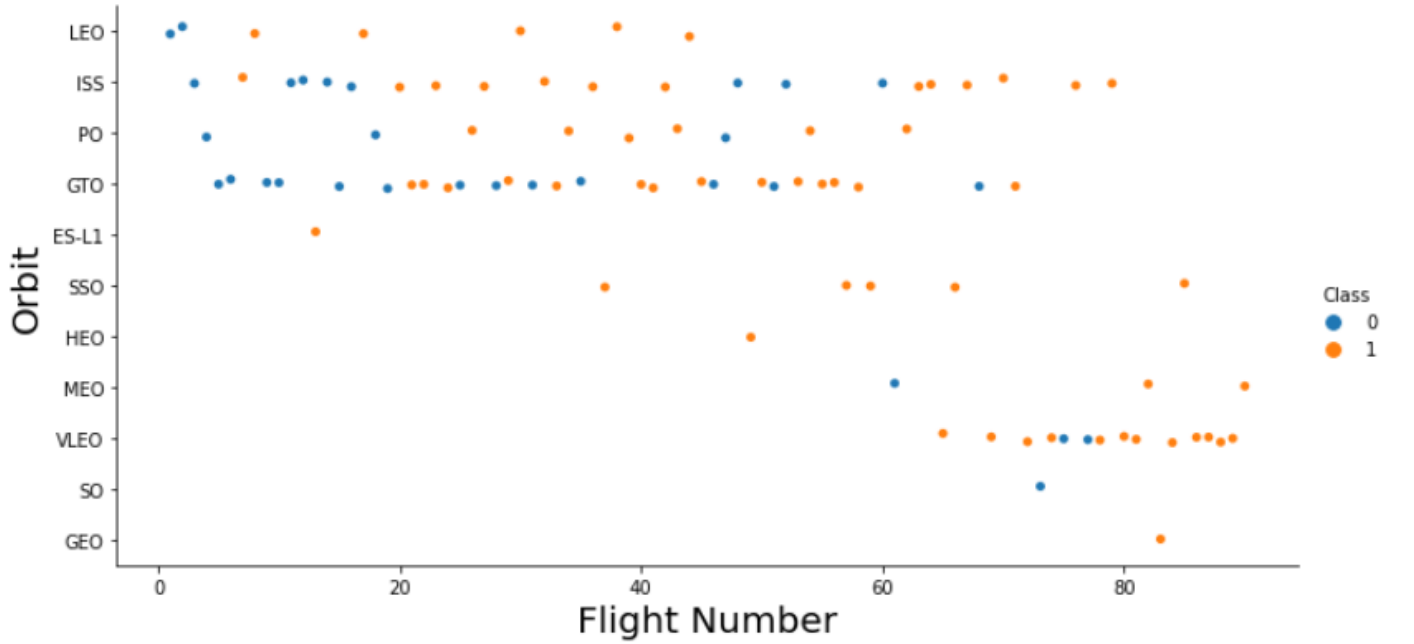
Flight Number vs. Orbit type

Show a scatter point of Flight number vs. Orbit type

Show the screenshot of the scatter plot with explanations

Launch Orbit preferences changed over Flight Number. Launch Outcome seems to correlate with this preference. SpaceX started with LEO orbits which saw moderate success LEO and returned to VLEO in recent launches SpaceX appears to perform better in lower orbits or Sun-synchronous orbits

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 2)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

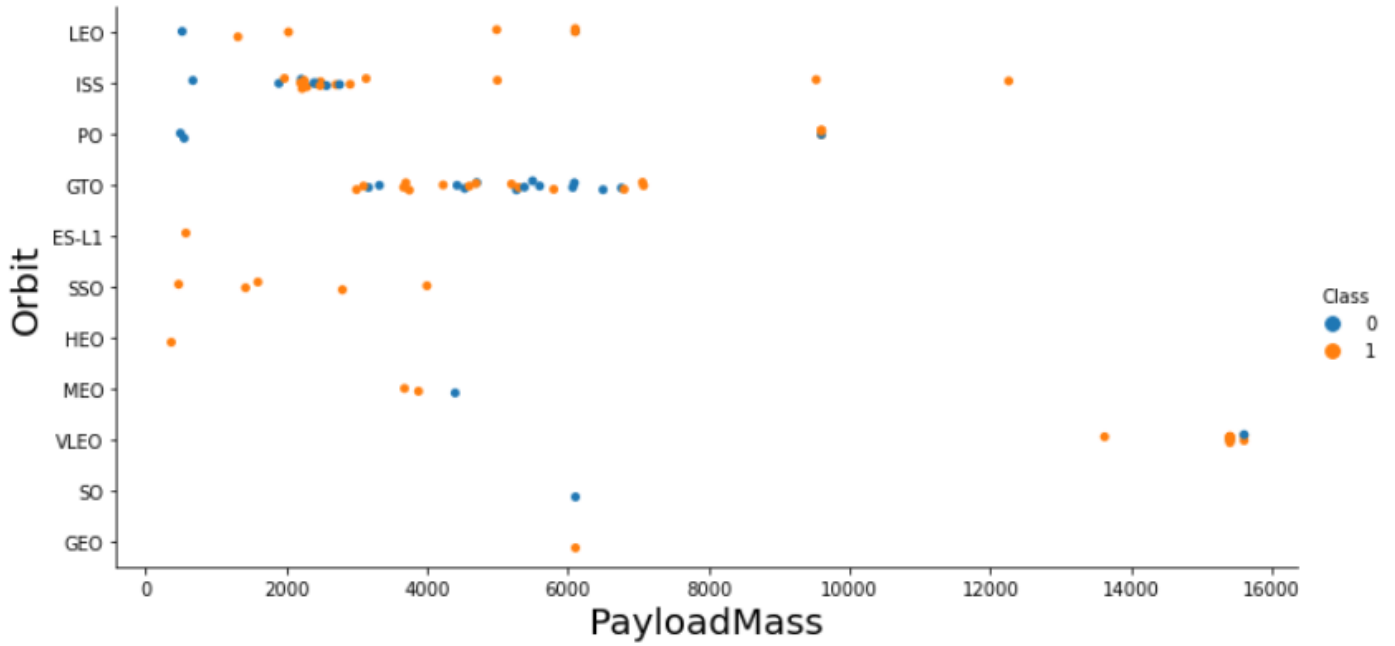


Payload vs. Orbit type

Show a scatter point of payload vs. orbit type

Show the screenshot of the scatter plot with explanations

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 2)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Payload mass seems to correlate with orbit

LEO and SSO seem to have relatively low payload mass

The other most successful orbit VLEO only has payload mass values in the higher end of the range

Launch success yearly trend

Show a line chart of yearly average success rate

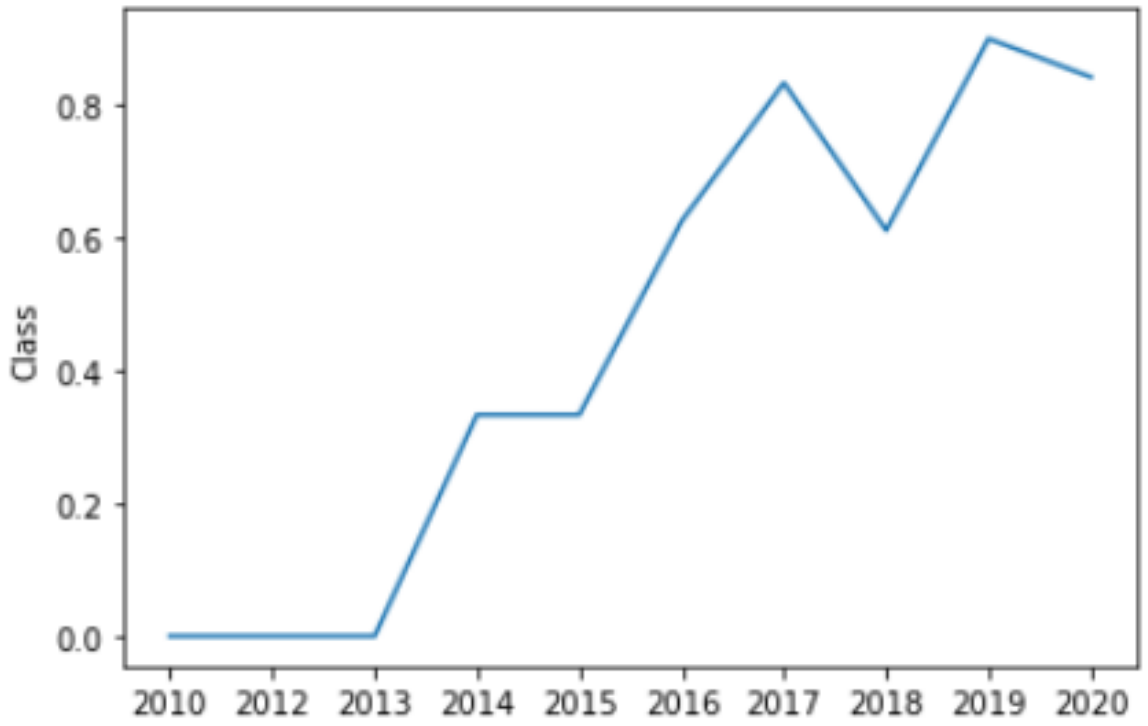
Show the screenshot of the scatter plot with explanations

Success generally increases over time since 2013 with a slight dip in 2018

Success in recent years at around 80%

```
sns.lineplot(data=df_new, x= np.unique(list_year)
                , y=df_new.groupby('year')['Class'].mean())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ecf3a702b0>
```



EDA with SQL

All launch site names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

Query unique launch site names from database.

CCAFS SLC-40 and CCAFSSLC-40 likely all represent the same launch site with data entry errors.

CCAFS LC-40 was the previous name.

Likely only 3 unique launch_site values:

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

```
In [9]: %sql SELECT DISTINCT(launch_site) FROM SPACEXTBL;
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81
Done.
```

Out[9]:

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

Launch site names begin with `CCA`

- Find all launch sites begin with `CCA`
- Present your query result with a short explanation here

First five entries in database with Launch Site name beginning with CCA.

```
%sql SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%'  LIMIT 5;
```

```
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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
- Present your query result with a short explanation here

This query sums the total payload mass in kg where NASA was the customer.

CRS stands for Commercial Resupply Services which indicates that these payloads were sent to the International Space Station (ISS).

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

sum_payload_mass_kg
45596

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

This query calculates the average payload mass or launches which used booster version F9 v1.1

Average payload mass of F9 1.1 is on the low end of our payload mass range

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

```
%sql SELECT AVG(payload_mass__kg_) FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%';
```

```
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.data
Done.
```

1
2534

First successful ground landing date

- Find the date when the first successful landing outcome in ground pad
- Present your query result with a short explanation here

This query returns the first successful ground pad landing date.

First ground pad landing wasn't until the end of 2015.

Successful landings in general appear starting 2014.

```
]: %sql SELECT DATE FROM SPACEXTBL WHERE landing_outcome LIKE 'Success (ground pad)' ORDER BY 1 ASC LIMIT 1;
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.clo
Done.
```

]:

DATE
2015-12-22

Successful drone ship landing with payload between 4000 and 6000

- List the names of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Present your query result with a short explanation here

This query returns the four booster versions that had successful drone ship landings and a payload mass between 4000 and 6000 noninclusively.

```
%sql SELECT booster_version,payload_mass__kg_ FROM SPACEXTBL
WHERE  payload_mass__kg_ BETWEEN 4000 AND 6000 and
landing_outcome LIKE '%drone%';
```

```
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8-86b520b8751
Done.
```

booster_version	payload_mass__kg_
F9 FT B1020	5271
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

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

This query returns a count of each mission outcome.

SpaceX appears to achieve its mission outcome nearly 99% of the time.

This means that most of the landing failures are intended.

Interestingly, one launch has an unclear payload status and unfortunately one failed in flight.

```
In [59]: %sql SELECT DISTINCT(mission_outcome) FROM SPACEXTBL;
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.da
Done.
Out[59]:
mission_outcome
Failure (in flight)
Success
Success (payload status unclear)

In [61]: %sql SELECT COUNT(mission_outcome) FROM SPACEXTBL;
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.da
Done.
Out[61]:
1
101

In [62]: %sql SELECT COUNT(mission_outcome) FROM SPACEXTBL WHERE mission_outcome LIKE 'Succ%';
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.da
Done.
Out[62]:
1
100

In [63]: %sql SELECT COUNT(mission_outcome) FROM SPACEXTBL WHERE mission_outcome LIKE 'Fail%';
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.da
Done.
Out[63]:
1
1
```

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

This query returns the booster versions that carried the highest payload mass of 15600 kg.

These booster versions are very similar and all are of the F9 B5 B10xx.x variety.

This likely indicates payload mass correlates with the booster version that is used.

```
%sql SELECT booster_version,payload_mass__kg_
FROM SPACEXTBL WHERE payload_mass__kg_ =
(SELECT MAX(payload_mass__kg_) FROM SPACEXTBL);
```

```
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8
Done.
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 launch records

- 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
- Present your query result with a short explanation here

This query returns the Month, Landing Outcome, Booster Version, Payload Mass (kg), and Launch site of 2015 launches where stage 1 failed to land on a drone ship.

There were two such occurrences.

```
In [92]: %sql select MONTH(DATE) as Month, landing_outcome,
         booster_version, launch_site
         FROM SPACEXTBL WHERE YEAR(DATE)='2015' and
         landing_outcome LIKE 'Failure%';

          * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8-86b
         Done.
```

Out[92]:

MONTH	landing_outcome	booster_version	launch_site
1	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
4	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank success count 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.
- Present your query result with a short explanation here

This query returns a list of successful landings and between 2010-06-04 and 2017-03-20 inclusively.

There are two types of successful landing outcomes: drone ship and ground pad landings.

There were 8 successful landings in total during this time period

In [77]:
```
%%sql
select landing_outcome, count(landing_outcome) from SPACEXTBL
where (DATE between '2010-06-04' and '2017-03-20') and landing_outcome like 'Success%'
group by landing_outcome
order by count(landing_outcome) desc
```

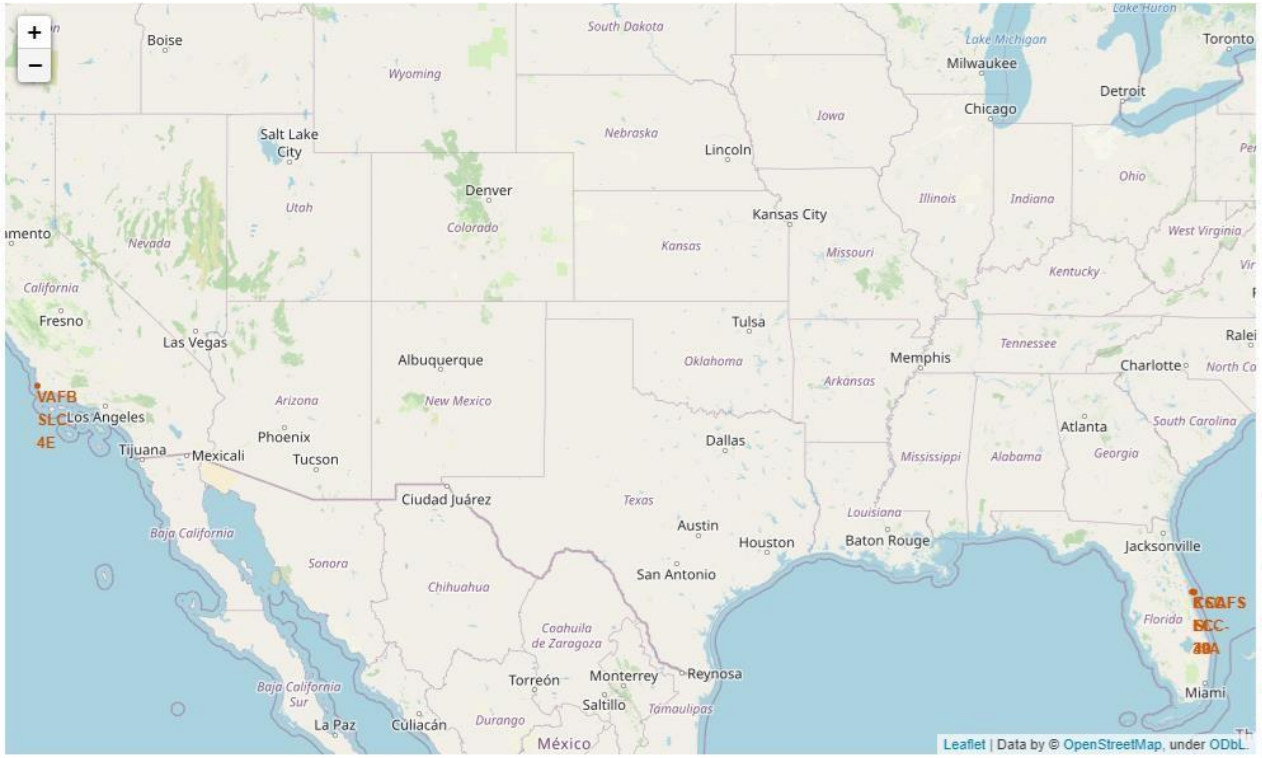
```
 * ibm_db_sa://tpg60270:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od8lcg.datal
Done.
```

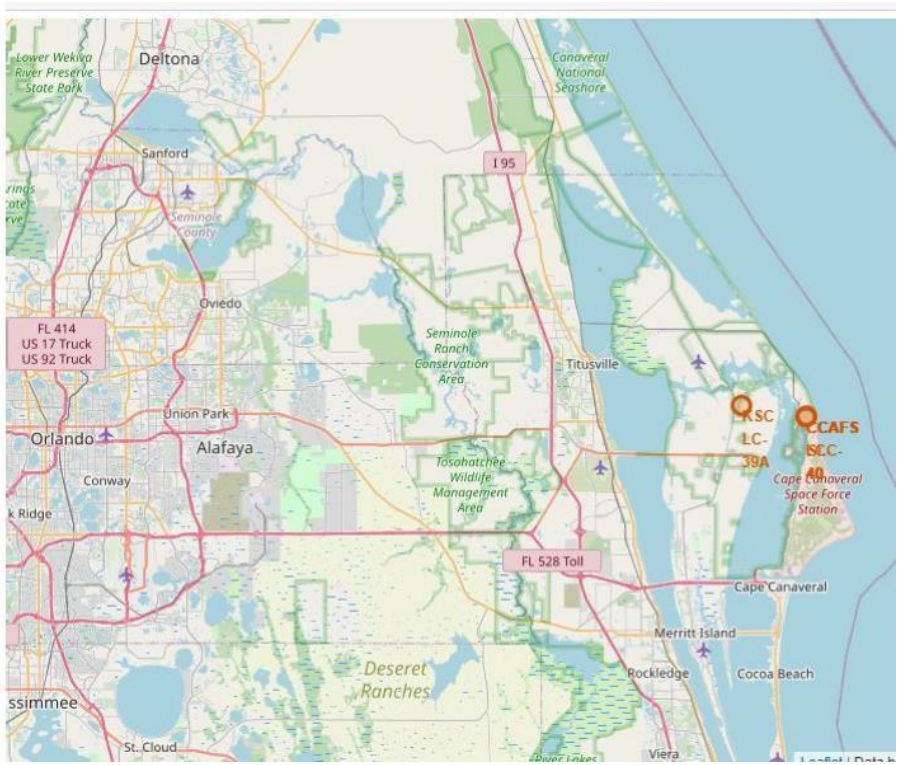
Out[77]:

landing_outcome	2
Success (drone ship)	5
Success (ground pad)	3

Interactive map with Folium

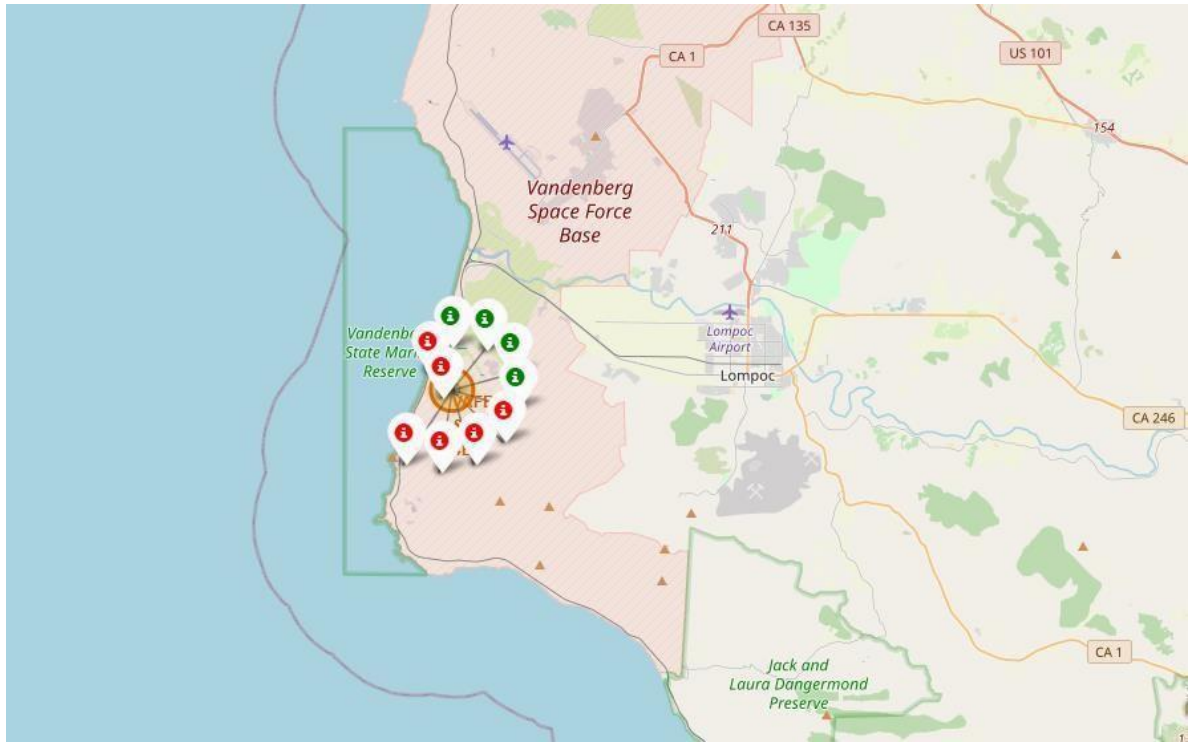
Launch Site Locations





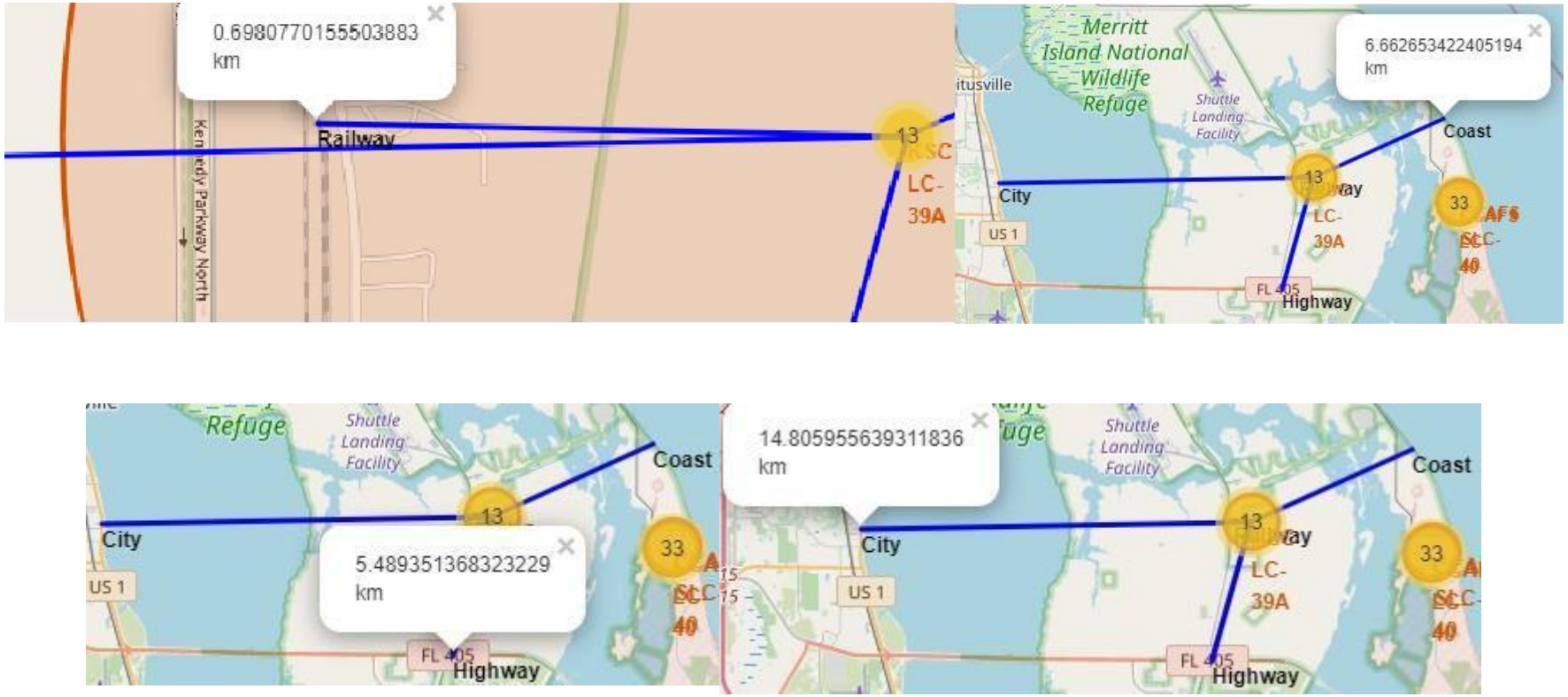
The left map shows all launch sites relative US map. The right map shows the two Florida launch sites since they are very close to each other. All launch sites are near the ocean.

Color Coded Launch Markers



Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed landing (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failed landings.

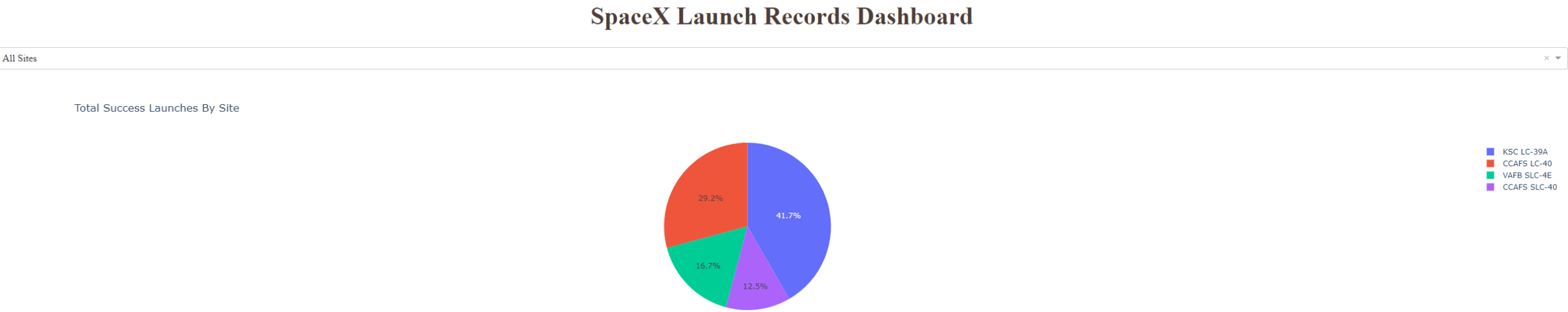
Key Locations Proximities



Using KSC LC-39A as an example, launch sites are very close to railways for large part and supply transportation. Launch sites are close to highways for human and supply transport. Launch sites are also close to coasts and relatively far from cities so that launch failures can land in the sea to avoid rockets falling on densely populated areas.

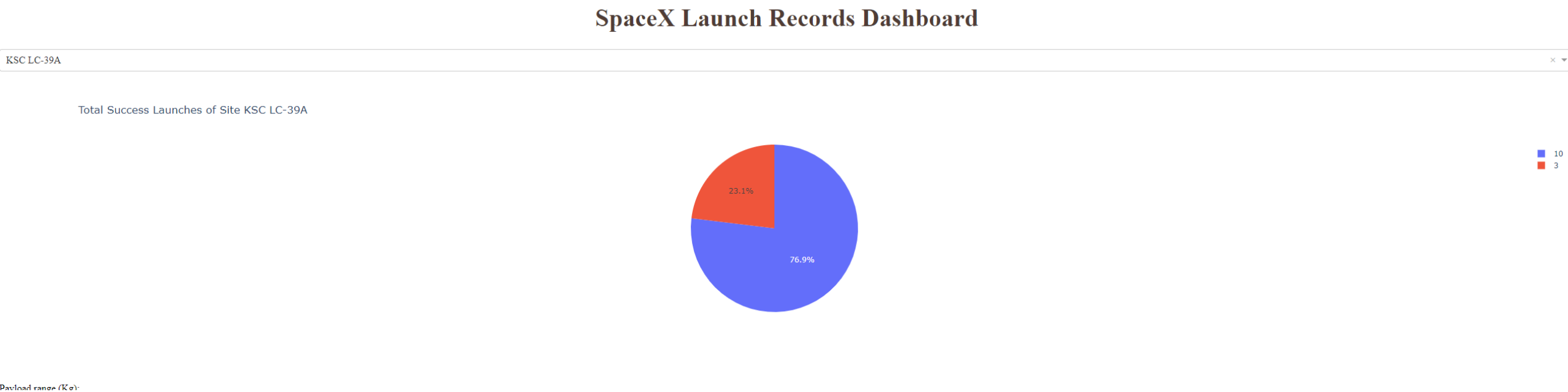
Build a Dashboard with Plotly Dash

Successful Launches for All Sites



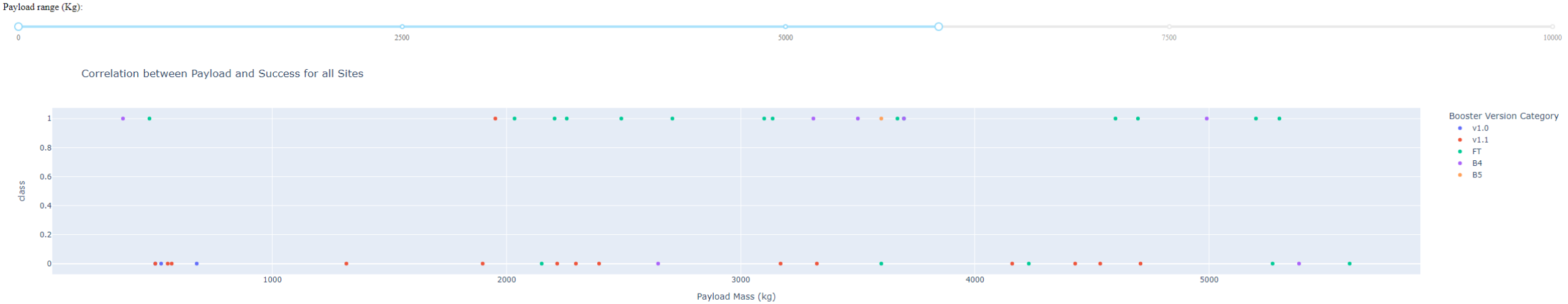
sites. CCAFS LC-40 is the old name of CCAFS SLC-40 so CCAFS and KSC have the same amount of successful landings, but a majority of the successful landings where performed before the name change. VAFB has the smallest share of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast.

Highest Success Rate for particular Site



KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

Correlation between Payload and Success Rate



Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size

Predictive analysis (Classification)

Classification Accuracy

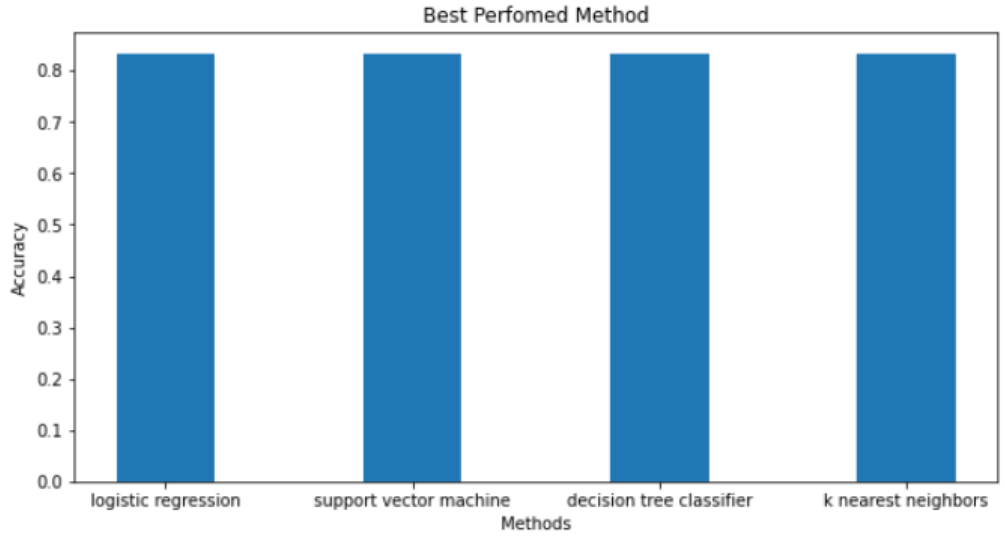
Visualize all the built model accuracy for all built models, in a barchart

Find which model has the highest classification accuracy

```
methods = ['logistic regression', 'support vector machine', 'decision tree classifier', 'k nearest neighbors']
acc = [0.8333333333333334, 0.8333333333333334, 0.8333333333333334, 0.8333333333333334]
fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(methods, acc,width = 0.4)

plt.xlabel("Methods")
plt.ylabel("Accuracy")
plt.title("Best Perfomed Method")
plt.show()
```



All models had virtually the same accuracy on the test set at 83.33% accuracy. It should be noted that test size is small at only sample size of 18.

This can cause large variance in accuracy results, such as those in Decision Tree Classifier model in repeated runs.

We likely need more data to determine the best model.

Confusion Matrix

Show the confusion matrix of the best perfming model with explanation

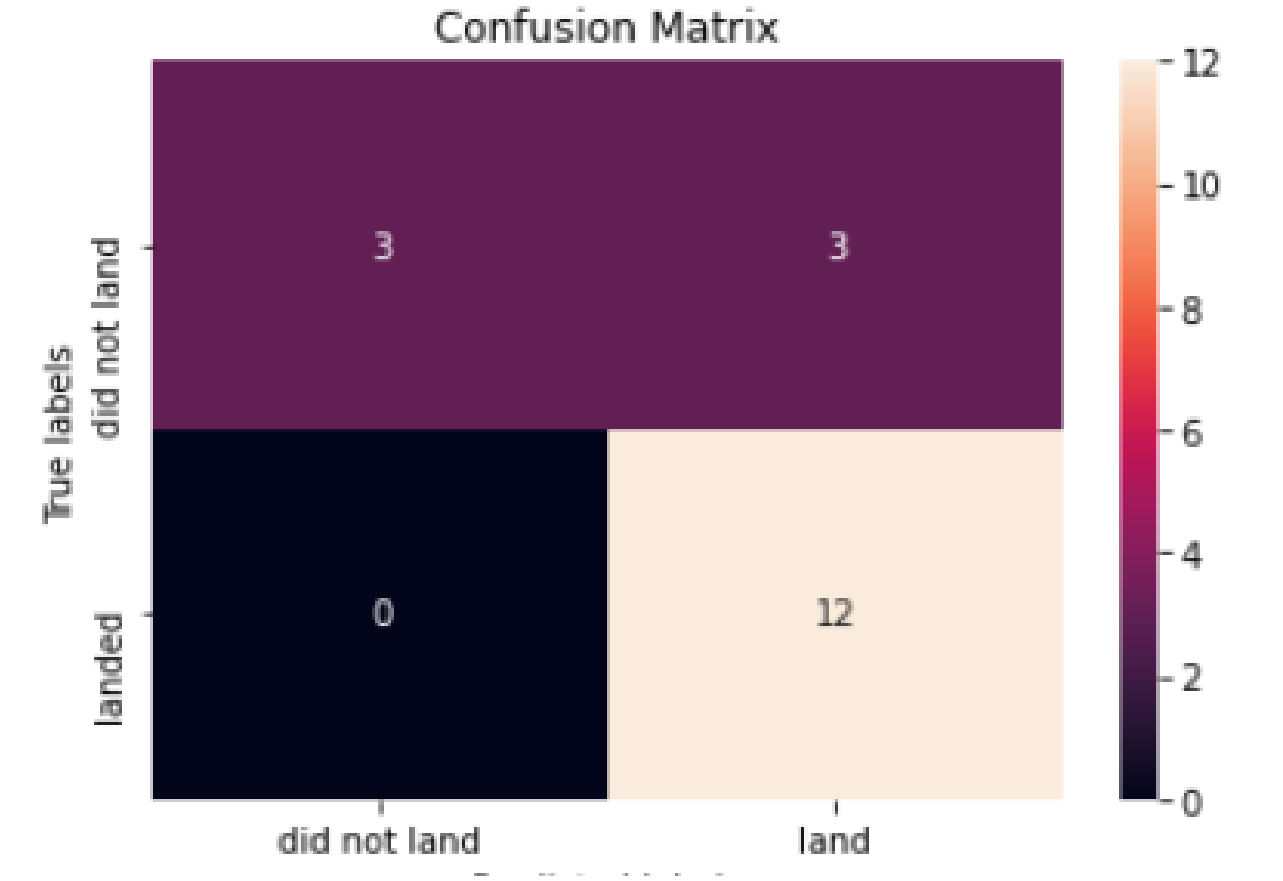
Correct predictions are on a diagonal from top left to bottom right.

Since all models performed the same for the test set, the confusion matrix is the same across all models. The models predicted 12 successful landings when the true label was successful landing.

The models predicted 3 unsuccessful landings when the true label was unsuccessful landing.

The models predicted 3 successful landings when the true label was unsuccessful landings (false positives). Our models over predict successful landings.

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



CONCLUSION

Our task: to develop a machine learning model for Space Y who wants to bid against SpaceX

The goal of model is to predict when Stage 1 will successfully land to save ~$100 million USD
Used data from a public SpaceX API and web scraping SpaceX Wikipedia page

Created data labels and stored data into a DB2 SQL database

Created a dashboard for visualization

We created a machine learning model with an accuracy of 83%

SpaceY can use this model to predict with relatively high accuracy whether a launch will have a successful Stage 1 landing before launch to determine whether the launch should be made or not

If possible more data should be collected to better determine the best machine learning model and improve accuracy

APPENDIX

- https://github.com/alimirzah/IBM_Data_Science_Certification