Winning Space Race with Data Science

Abhivarma Birru
March 3rd 2024

Outline

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

Executive Summary

- 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.

- 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

Background

- The commercial space race is going on
- Space X has best pricing ($62 million vs. $165 million USD)
- Largely due to ability to recover part of rocket (Stage 1)
- Space Y wants to compete with Space X

Problem:

- The company Space Y aims to train a machine learning model to predict successful Stage 1 recovery

Section 1

Methodology

Methodology

- Data collection methodology:
  - We have collected data from SpaceX public API and we have scraped data from SpaceX Wikipedia page.
- Perform data wrangling
  - The data was preprocessed to classify true landings as successful and false landings as unsuccessful
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We have trained 4 models Linear, Logistic regression, Decision Tree and KNN
  - Using GRIDCV, we have analyzed cross validation on the four models
  - These models are compared to find the best model

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.

Space X API Data Columns:

FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

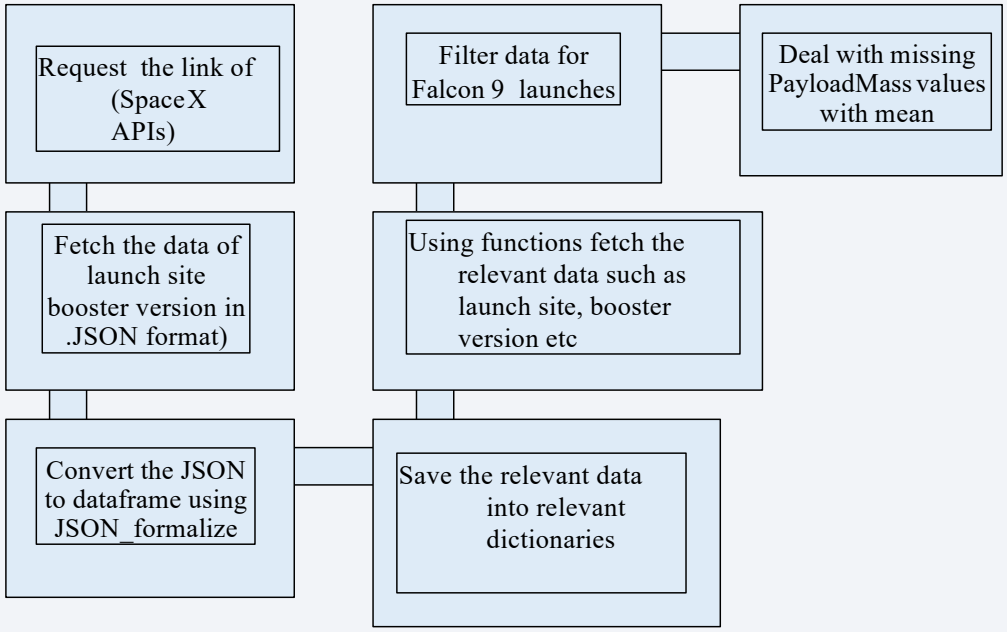
Wikipedia Webscrape Data Columns:

Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection – SpaceX API

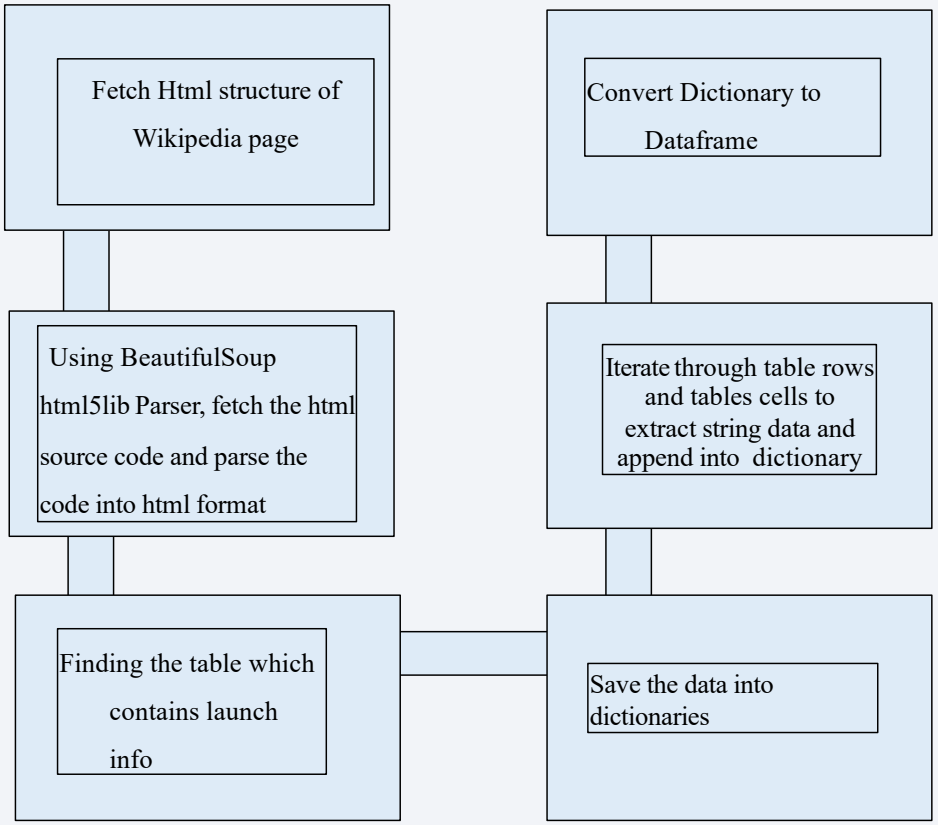
- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- GitHub URL : https://github.com/ABV14/Coursera/blob/main/Applied%20Capstone%20Project/Week%201/Data%20Collection%20API/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- GitHub URL : https://github.com/ABV14/Coursera/blob/main/Applied%20Capstone%20Project/Week%201/Data%20Collection%20with%20Web%20Scraping/jupyter-labs-webscraping.ipynb



Data Wrangling

- We have created a training label with landing outcomes where successful = 1 & failure = 0.
- The outcome has two components: ‘Mission Outcome’ ‘Landing Location’
- We have created a 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/ABV14/Coursera/blob/main/Applied%20Capstone%20Project/Week%201/Data%20Wrangling/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

The Exploratory Data Analysis is performed on features such as 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/ABV14/Coursera/blob/main/Applied%20Capstone%20Project/Week%202/EDA%20with%20Visualization/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

We have used SQL queries using SQL Python integration.

The queries are used to get a better understanding of the dataset.

We have used sql queries to get information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

GitHub url:

https://github.com/ABV14/Coursera/blob/main/Applied%20Capstone%20Project/Week%202/EDA%20with%20SQL/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Using the Folium maps we have marked Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.

This creates visualization insights to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.

GitHub url:

https://github.com/ABV14/Coursera/tree/main/Applied%20Capstone%20Project/Week%203/Interactive%20Visual%20Analytics%20with%20Folium%20lab

Build a Dashboard with Plotly Dash

Created a Dashboard which includes a pie chart and a scatter plot.

This 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.

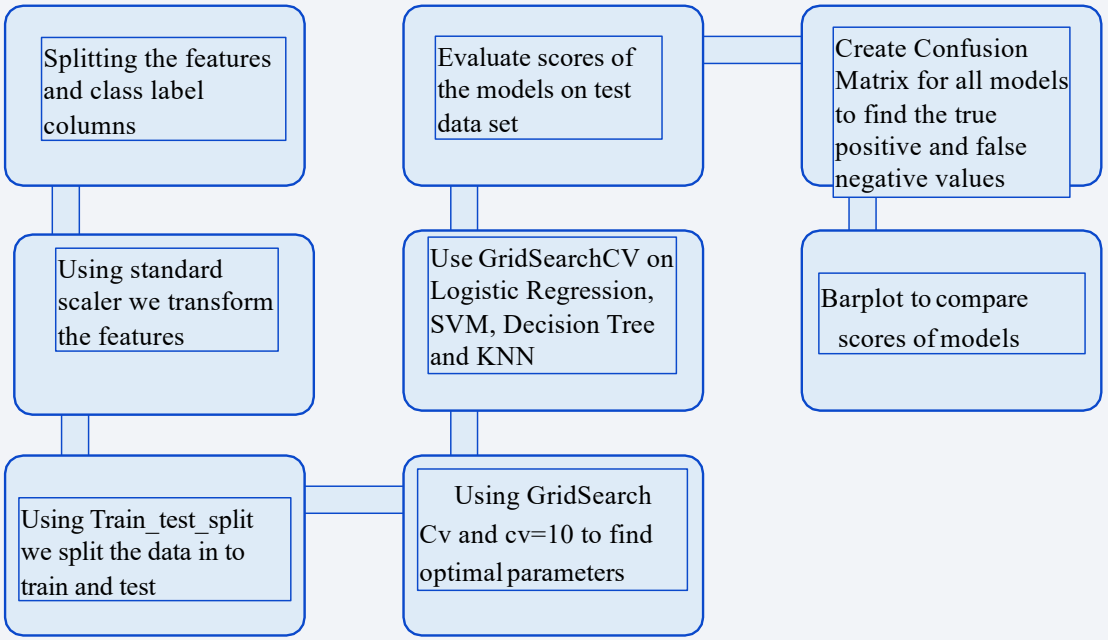
The scatter plot can help us see how success varies across launch sites, payload mass, and booster version category.

GitHub url:

https://github.com/ABV14/Coursera/tree/main/Applied%20Capstone%20Project/Week%203/Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash

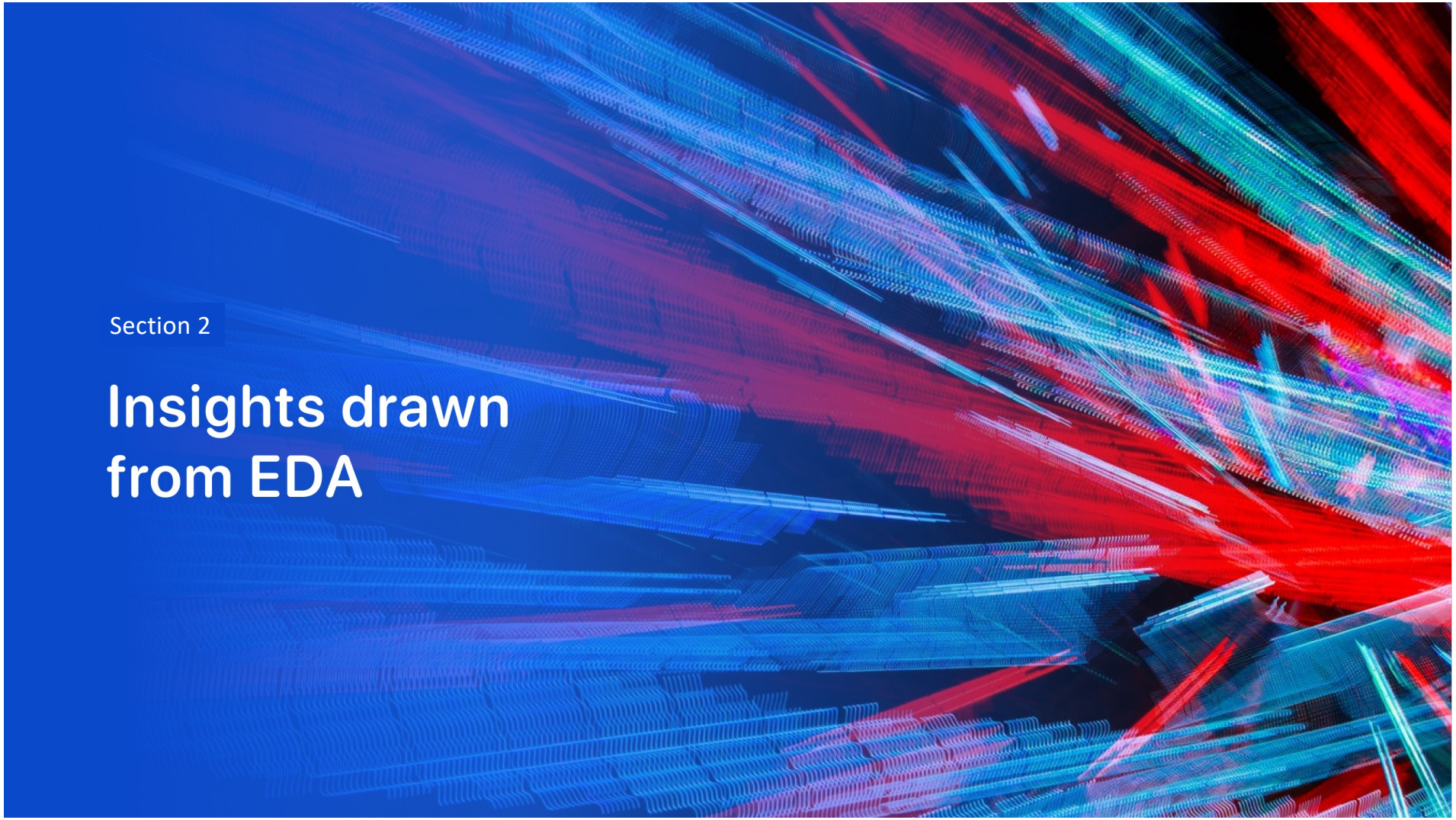
Predictive Analysis (Classification)

GitHub url:
https://github.com/ABV14/Coursera/blob/main/Applied%20Capstone%20Project/Week%204/Machine%20Learning%20Prediction/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

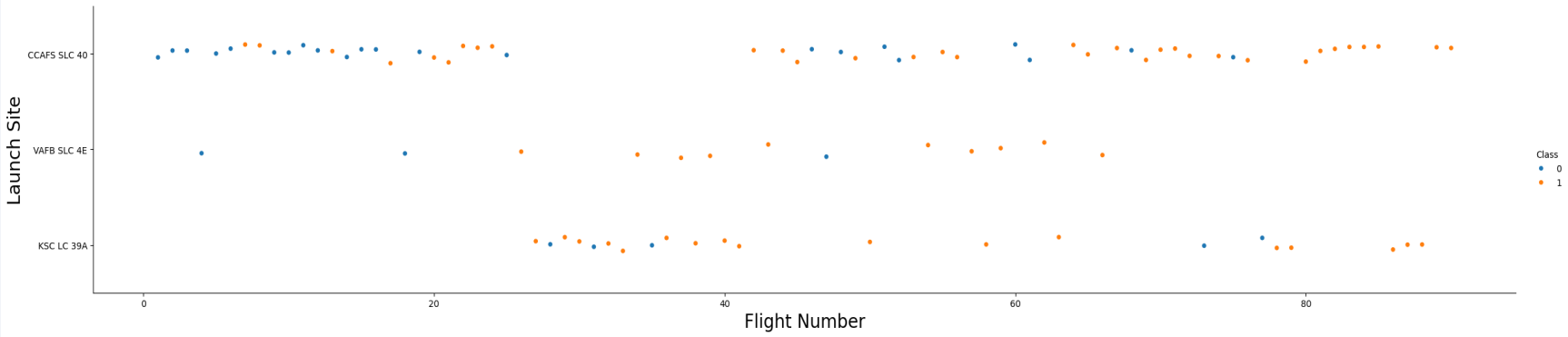


Results

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


Section 2
Insights drawn
from EDA

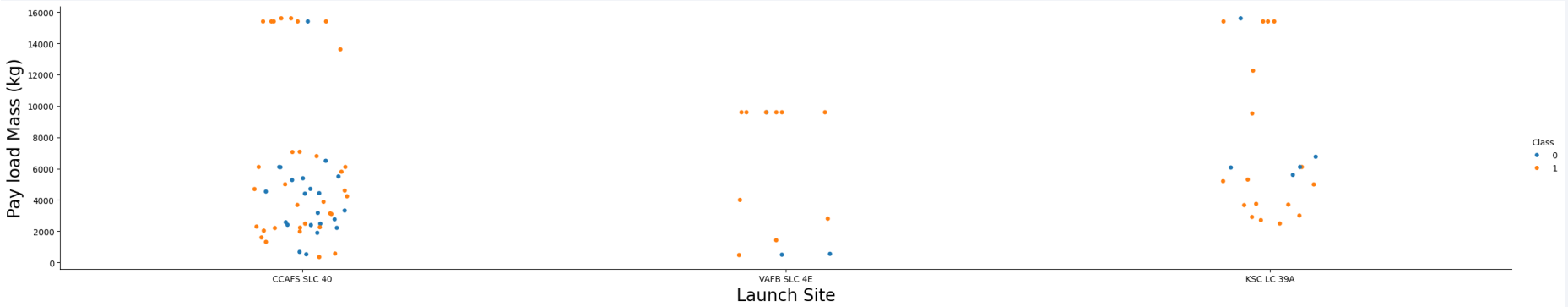
Flight Number vs. Launch Site



- Here Blue dots are unsuccessful launches and red orange dots are successful launches

- The graph suggests that an increase in success rate over time. There are= around flight 20 which significantly increased success rate. CCAFS is the main launch site with high volume as suggested from the graph

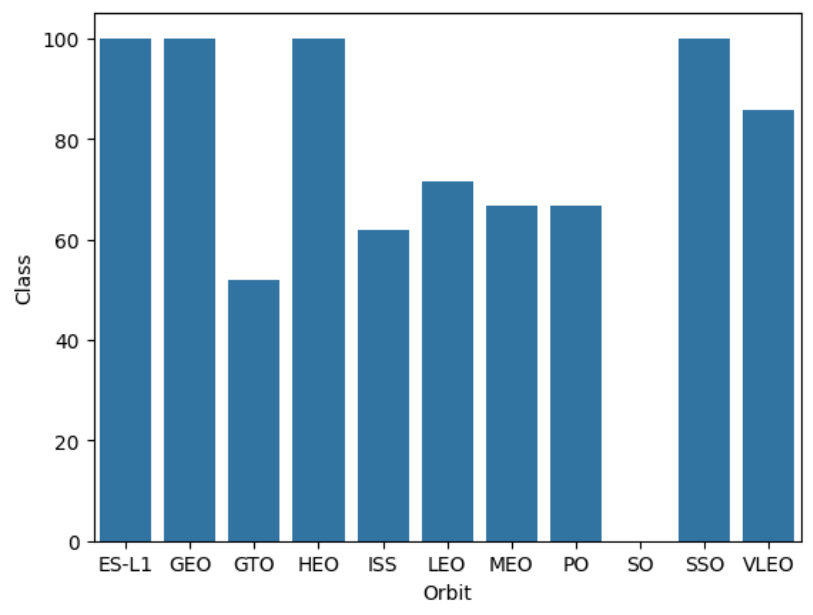
Payload vs. Launch Site



- Here Blue dots are unsuccessful launches and red orange dots are successful launches

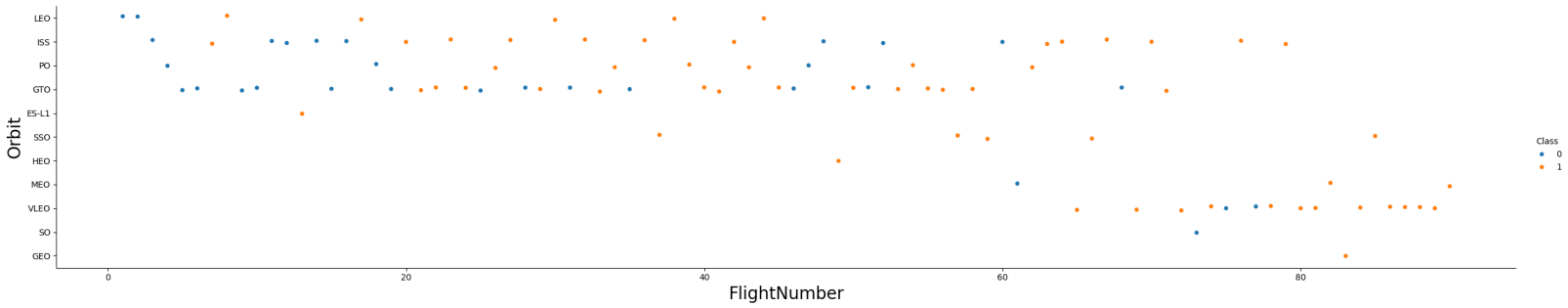
- The graph suggests that major payload is mostly between 0-6000kg and 16000kg payload mass launches are taken place in CCAFS and KSC launch sites.

Success Rate vs. Orbit Type



- ES-L1, GEO, HEO , SSO have 100% success rate
- VLEO has decent success rate of around 80%
- SO has 0% success rate
- GTO has around 50% success rate

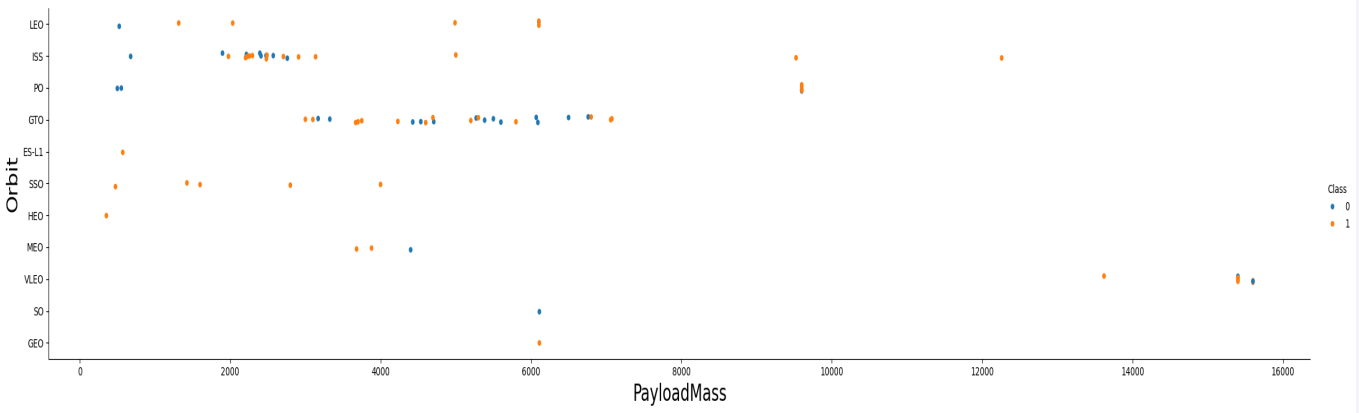
Flight Number vs. Orbit Type



Here Blue dots are unsuccessful launches and red orange dots are successful launches

- The Launch Orbits preferences changes for different Flight Number.
- Launch Outcomes have changed 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

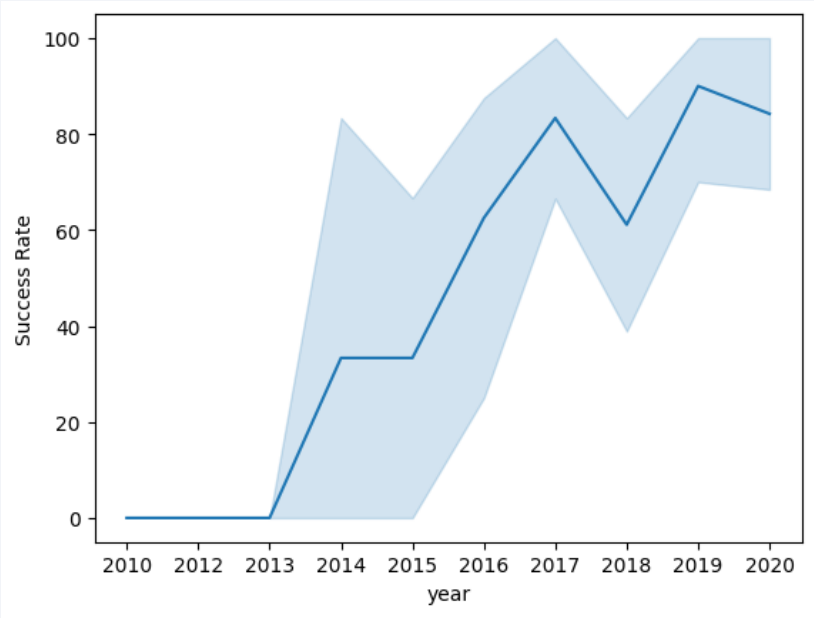
Payload vs. Orbit Type



Here Blue dots are unsuccessful launches and red orange dots are successful launches

- Payload mass values did correlate with orbit
- LEO and SSO have relatively low payload mass
- The other most successful orbit VLEO only has payload mass values of higher range

Launch Success Yearly Trend



- The Success rate increased from 2013 with a slight dip in 2018
- Success in recent years at around 80%

All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

In [9]:
```
%sql select distinct("Launch_Site") from SPACEXTBL
```

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

Out[9]:

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

- This query fetches distinct launch site names from database.
- CCAFS SLC-40 and CCAFSSLC-40 represent same launch site.
- The three launch sites are:

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

Launch Site Names Begin with 'CCA'

Task 2

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

In [10]:
```
%sql select "Launch_Site" from SPACEXTBL WHERE "Launch_Site" LIKE "CCA%" limit 5;
```

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

Out[10]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Fetch the first five entries of the launchsite whose name starts with CCA

Total Payload Mass

Task 3

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

```
[20]: %sql select sum(payload_mass__kg_) as "Total Payload of NASA(CRS)" from SPACEXTBL where customer like 'NASA (CRS)'
```

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

[20]:

Total Payload of NASA(CRS)
45596

This query adds all the payload mass and shows the total payload mass in kg where NASA is the customer.

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

Average Payload Mass by F9 v1.1

Task 4

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

```
[12]: %sql select avg("PAYLOAD_MASS_KG_") as "Avg payload of F9 v1.1" from SPACEXTBL where "Booster_Version" like 'F9 v1.1'
```

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

[12]:

Avg payload of F9 v1.1
2928.4

This query returns the average payload mass of the launches which used booster version F9 v1.1

Average payload mass of F9 1.1 is on the lower range payload mass

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
[13]: %sql select min(date) as Date from SPACEXTBL where "Mission_Outcome" like 'Success'
```

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

[13]:

Date
2010-06-04

This query returns the first successful ground pad landing date.

Successful landings in general appear starting 2014.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

[14]:
```
%%sql select "Booster_Version" from SPACEXTBL
where ("Mission_Outcome" like 'Success') AND ("PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000) AND ("Landing_Outcome" like 'Success (drone ship)')
```

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

[14]:

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

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

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
[15]: %sql SELECT "Mission_Outcome", count(*) as Count FROM SPACEXTBL GROUP by "Mission_Outcome" ORDER BY "Mission_Outcome"
```

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

[15]:

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

This query returns a count of each mission outcome grouped by mission outcome.

SpaceX achieved its mission outcome 98% of the launches.

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

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[22]: %sql select "booster_version" from SPACEXTBL where "PAYLOAD_MASS__KG_"=(select max("PAYLOAD_MASS__KG_") from SPACEXTBL)
 * sqlite:///my_data1.db
Done.
```

[22]:

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

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

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

2015 Launch Records

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.

```
[23]: %%sql

select  substr(Date, 6,2) as "Month", "Landing_Outcome", "Booster_Version", "Launch_Site" from SPACEXTBL
where Date like "2015%" AND "Landing_Outcome" like 'Failure (drone ship)'
```

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

[23]:

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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.

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

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.

[24]:
```
%sql select "Landing_Outcome", count(*) as count from SPACEXTBL where Date >= '2010-06-04' AND Date <= '2017-03-20' GROUP by "Landing_Outcome" ORDER BY count Desc
```

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

[24]:

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

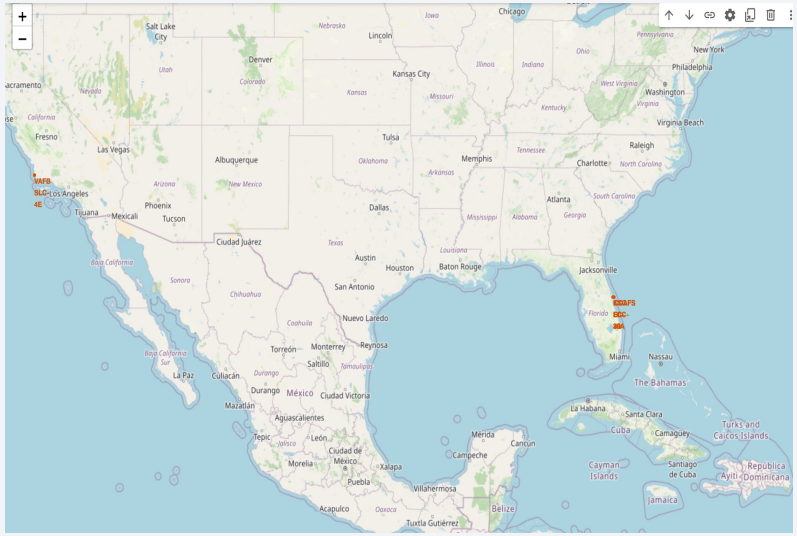
This query returns a list of successful landings and between 2010-06-04 and 2017-03-20.

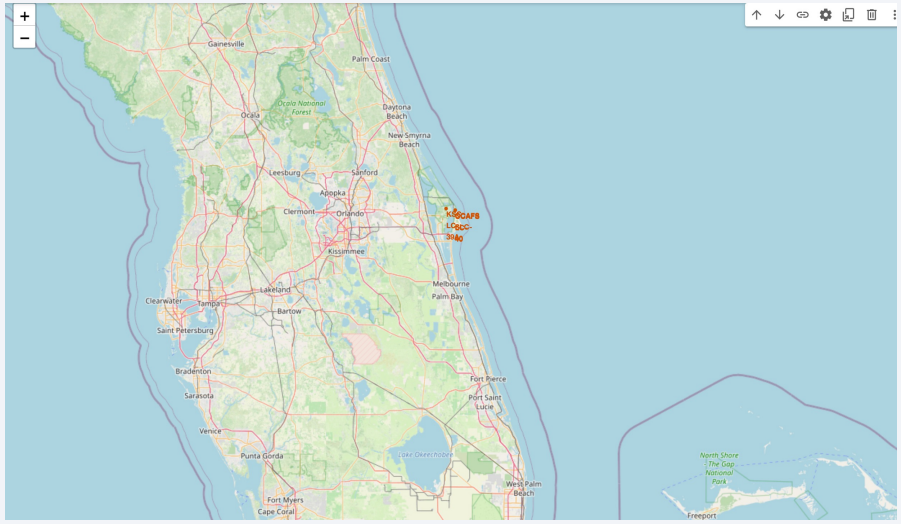
There are two types of successful landing outcomes: drone ship and ground pad landings.

There were 8 successful landings in during 2010-06-04 and 2017-03-20


Section 3
Launch Sites
Proximities Analysis

Launch Sites



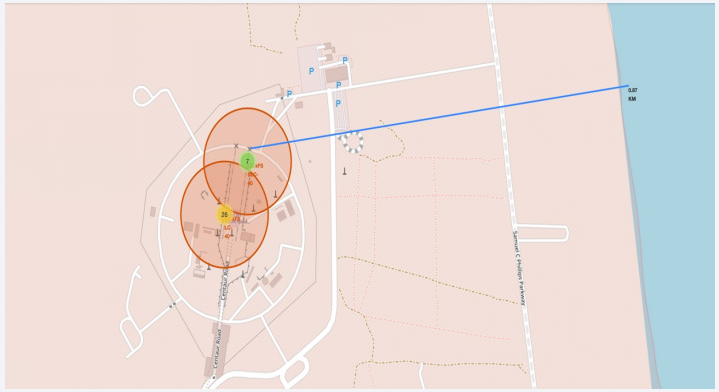


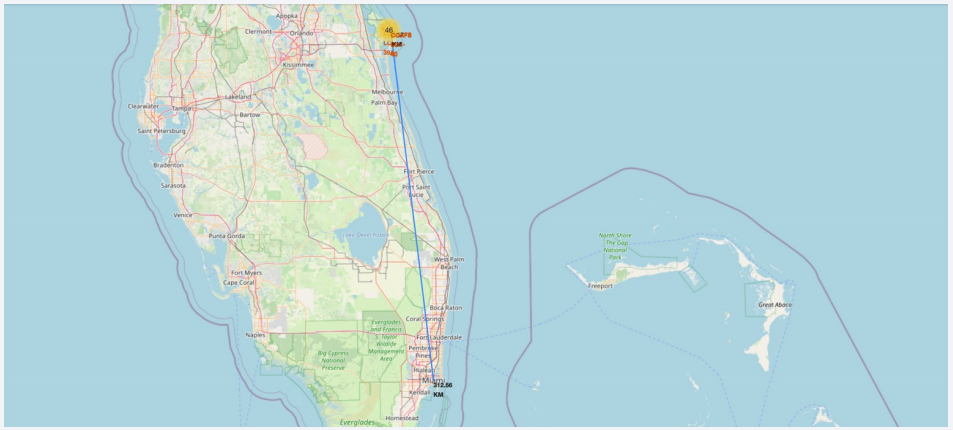
The left map shows us the launch sites according to the map of usa. The right map shows the launch sites in Florida. All launch sites are placed near to ocean.

LAUNCH MARKERS

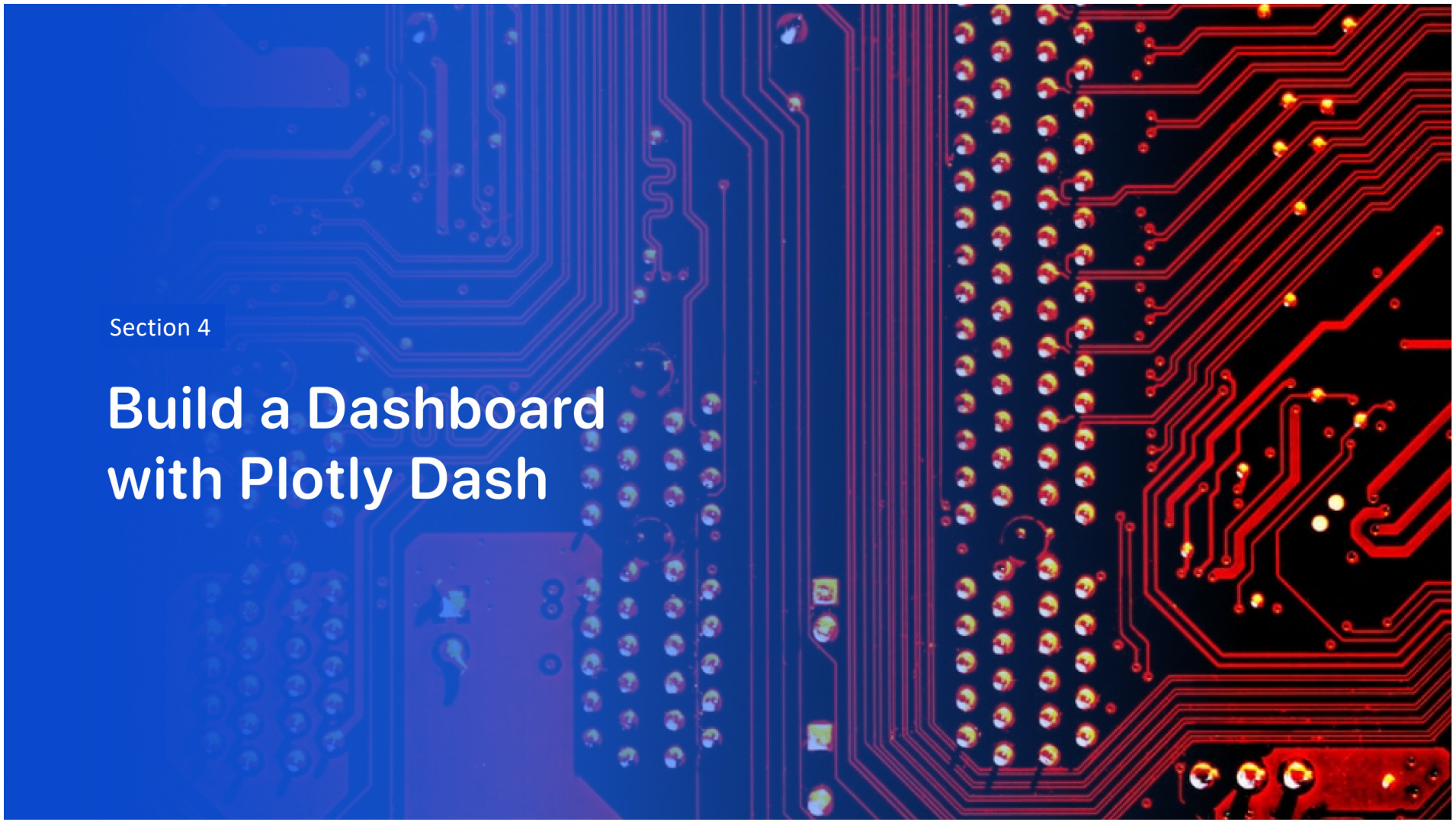
- On Folium map upon click on the cluster we can display each successful landing (green icon) and failed landing (red icon). In this example KSC LC 39A shows 10 successful landings and 3 failed landings.

Key Location Proximities

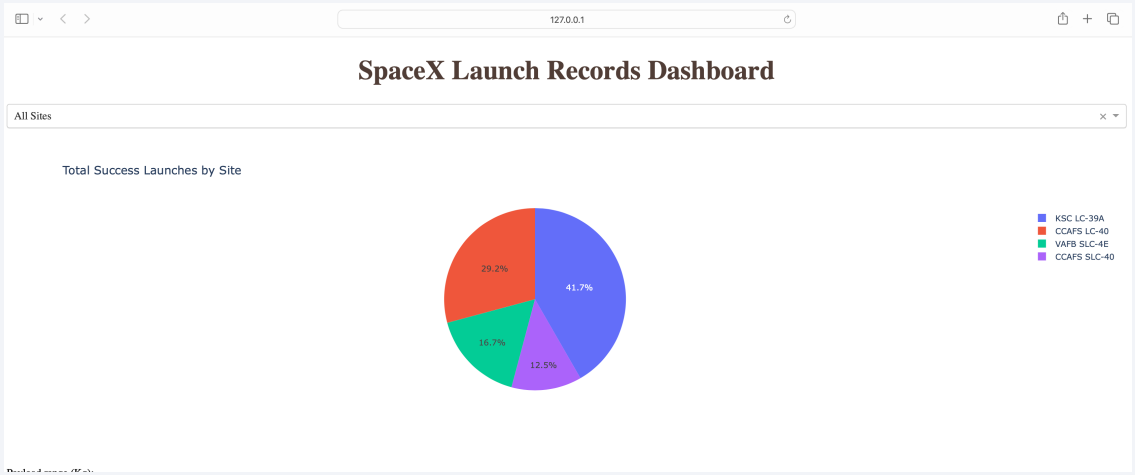




- Using CCAFS SLC-40 as an example, launch sites are very close to coasts and relatively far from cities so that launch failures can land in the sea to avoid rockets falling on densely populated areas.

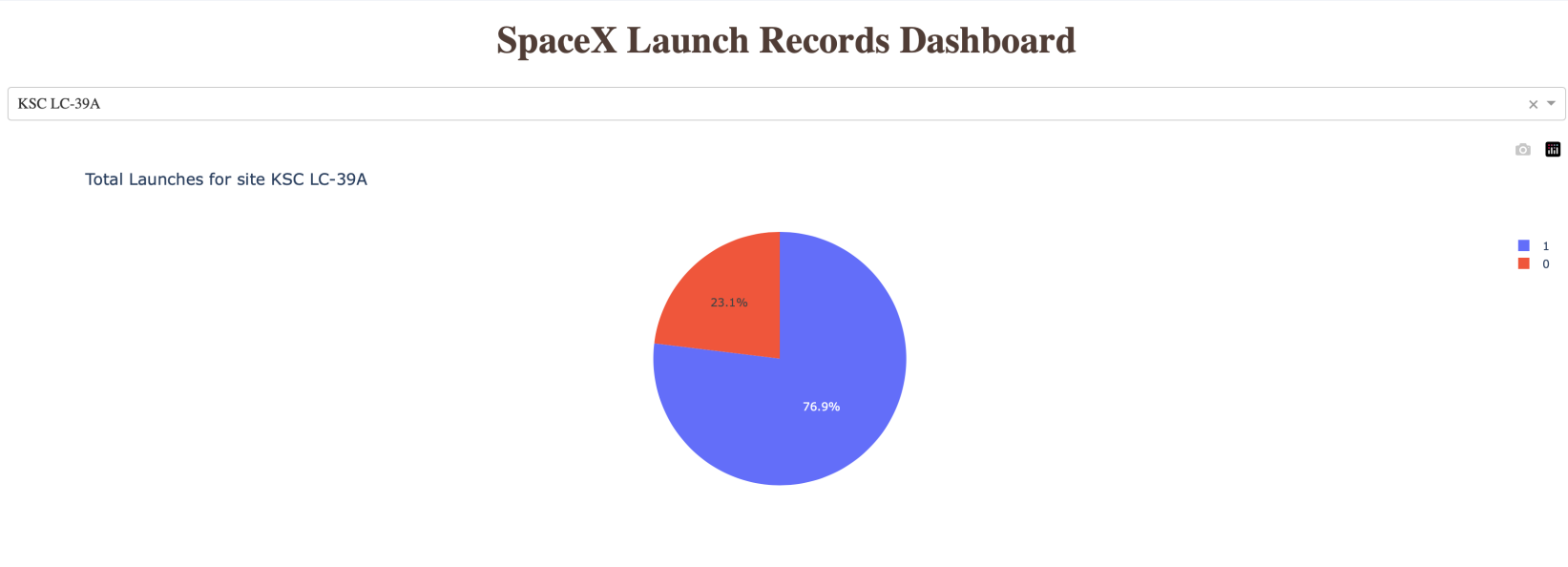

Section 4
Build a Dashboard with Plotly Dash

Successful Launches



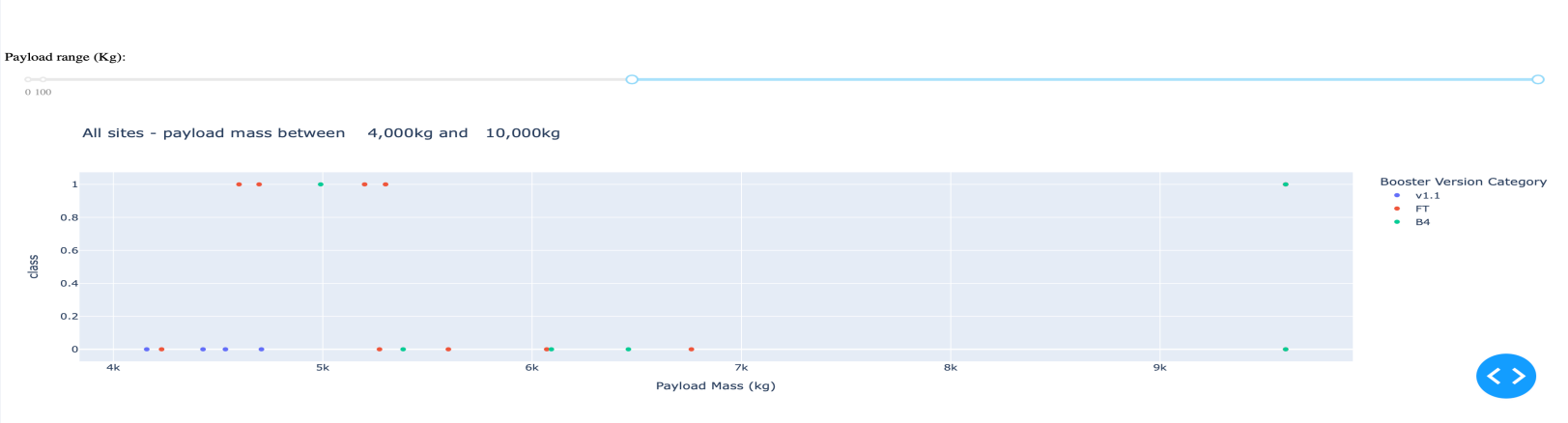
- This is the distribution of successful landings across all launch sites. CCAFS LC-40 and CCAFS SLC-40 are similar. KSC has a majority of the successful landings. VAFB has the smallest share of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast.

Highest Success of Rate of KSC LC-39A



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

PAYLOAD VS BOOSTER VS SUCCESS RATE

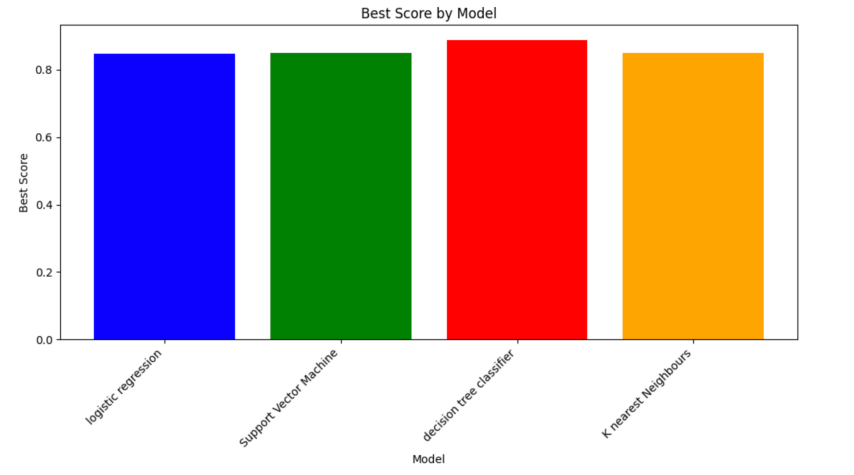


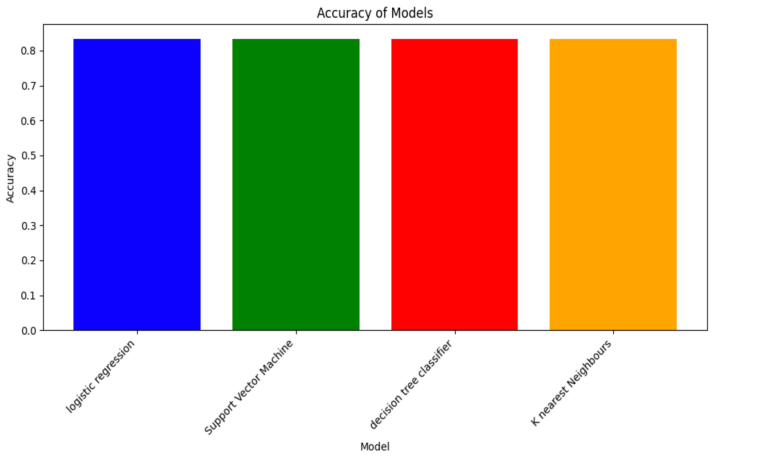
- Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600.
- Class indicates 1 for successful landing and 0 for failure.
- Scatter plot also accounts for booster version category in color and number of launches in point size. In this particular range of 4000KG – 10000KGS, we find the FT version has high success rate

Section 5

Predictive Analysis (Classification)

Classification Accuracy



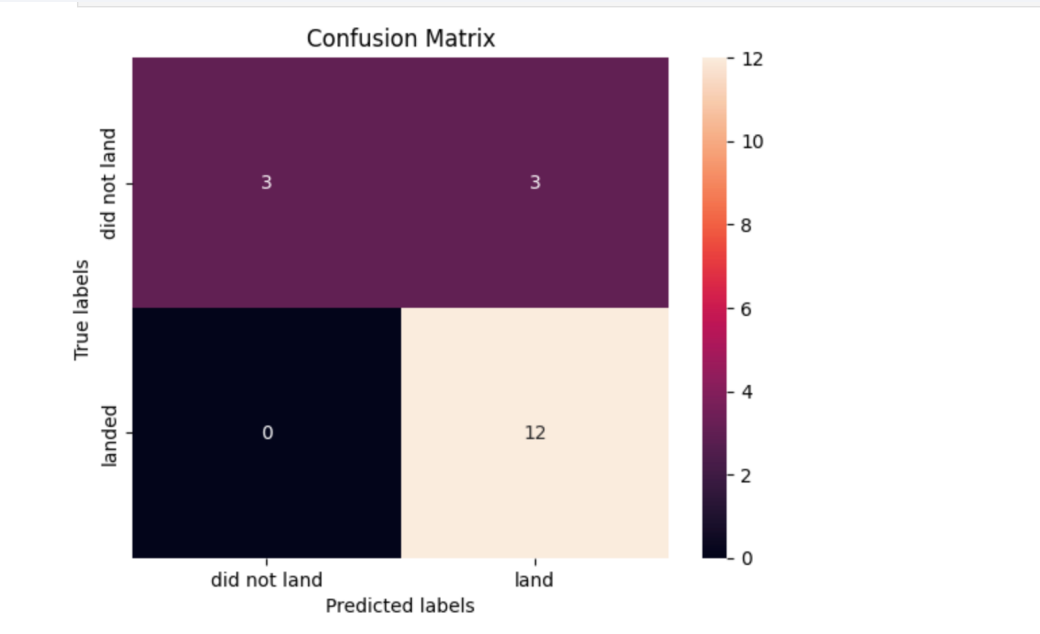


All models have the same accuracy on the test set at 83.33% accuracy.

Based on the best scores, the Decision tree classifier performed well compared to other models.

Much more data is need for final determination of best model

Confusion Matrix



- The confusion matrix is the same across all models. The models predicted 12 successful landings as we considered the true label as 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).

Conclusions

- We have developed a machine learning model for Space Y who wants to bid against SpaceX.
- The goal of model is to predict the Stage 1 success rate for landing to save ~$100 million USD.
- Collected data from a public SpaceX API and web scraping SpaceX Wikipedia page.
- Created data labels and stored data into a SQL database.
- Created a dashboard for visualization of key features of the dataset.
- Developed a machine learning model with an accuracy of 83%.
- Allon Mask of SpaceY can use this model to predict with relatively high accuracy whether a launch will have a successful Stage 1 landing .
- This helps before launch to decide whether the launch should be made or not.
- More data should be collected to determine the best machine learning model and improve accuracy.

Appendix

GitHub repository url:

https://github.com/ABV14/Coursera/tree/main/Applied%20Capstone%20Project

Special Thanks to All Instructors:

https://www.coursera.org/professional-certificates/ibm-data-science?#instructors

Thank you!