IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Taaha Saleem Bajwa
06 Nov, 2022

Outline

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

Executive Summary

- SpaceX offers cheap rocket launches compared to competitors as it has developed technology to reuse and land the first stage.
- By predicting if SpaceX rocket will land or not, we can bid against SpaceX for rocket launches
- SpaceX success rate has increased impressively from 2013 to 2020
- Using classification algorithms, we can predict with 83.33% accuracy if the SpaceX rocket will land or not

Introduction

- SpaceX has developed technology allowing it to reuse the first stage after a launch.
- This results in cheaper space launches and allows SpaceX to provide low cost rocket launches to customers.
- If we can successfully predict that if SpaceX will land the first stage we can predict the cost of the launch.
- Our prediction can allow another company to estimate cost of launch and bid against the launch.

Section
1

Methodology

Methodology

Executive Summary

- Data collection was done from SpaceX API and webscraping of SpaceX wikipedia page
- Data wrangling was done by removing missing values and typecasting incorrect datatypes. A column having class 1 or 0 was added.
- Relationship of target variable with independent variables were visualised and analysed in python. Insights were also obtained by passing queries to data in SQL.
- Launch Site locations were analysed in Folium and Success rate for each launch site was analysed through interactive plotly dashboard
- Logistic regression, SVM, Decision Tree and KNN Classifiers were built and tuned to accurately predict landing success (83% accuracy)

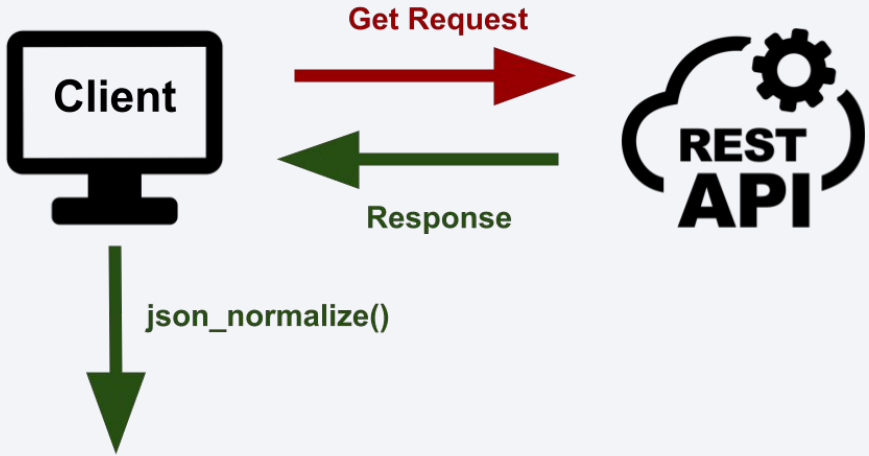
Data Collection

- Data was collected from the following sources

1. SpaceX API
2. Web Scraping from Wikipedia

Data Collection – SpaceX API

- API Endpoint used was https://api.spacexdata.com/v4/launches/past
- Get request was used to get a json response which was converted into pandas dataframe
- Github URL of notebook is https://github.com/taaha/SpaceX-Capstone-Project/blob/master/Data_Collection_from_API.ipynb

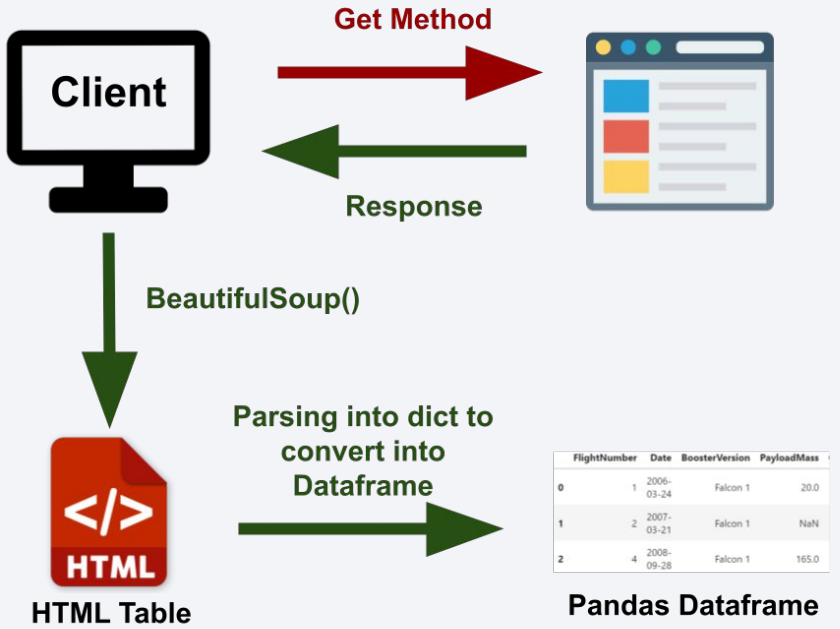


	FlightNumber	Date	BoosterVersion	PayloadMass
0	1	2006-03-24	Falcon 1	20.0
1	2	2007-03-21	Falcon 1	NaN
2	4	2008-09-28	Falcon 1	165.0

Pandas Dataframe

Data Collection - Scraping

- Webpage used for scraping was https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Launch records were scraped from webpage in form of html tables which was then converted into pandas dataframe
- Github url of notebook is https://github.com/taaha/SpaceX-Capstone-Project/blob/master/Data_Collection_From_Web_Scraping.ipynb



Data Wrangling

- Data was first checked for missing values and incorrect data types
- Different type of landings were divided into two classes and class column was added
    a. Successful Landing (Class=1)
    b. Failed Landing (Class=0)
- GitHub URL of notebook is https://github.com/taaha/SpaceX-Capstone-Project/blob/master/Data_Wrangling.ipynb

Raw Data

Removing missing values
Correcting data types

Organising data into desired form

	FlightNumber	Date	BoosterVersion	PayloadMass
0	1	2010-06-04	Falcon 9	6104.959412
1	2	2012-05-22	Falcon 9	525.000000
2	3	2013-03-01	Falcon 9	677.000000

Clean Dataframe

EDA with Data Visualization

- Relationship between of target variable with different independent variables was analysed visually to get useful insights.
- Charts plotted were
    - Categorical scatter plot of Flight Number vs Launch Site
    - Categorical scatter plot of Payload vs Launch Site
    - Bar plot of Success Rate vs Orbit type
    - Categorical scatter plot of Flight Number vs Orbit type
    - Categorical scatter plot of Payload vs Orbit type
    - Line plot of Success rate yearly trend
- Github url of notebook - https://github.com/taaha/SpaceX-Capstone-Project/blob/master/EDA_with_Python_Visualization.ipynb

EDA with SQL

- SQL queries performed were
  - Name of all launch sites
  - Records in which launch sites name begin with ‘CCA’
  - Total Payload Mass
  - Average Payload Mass for F9 v1.1 booster
  - First Successful ground landing date
  - Successful drone ship landing with payload mass between 4000 and 6000
  - Number of successful and failed mission outcomes
  - Boosters which carried maximum payload
  - 2015 Launch Records
  - Landing records between 06/04/2010 and 20/03/2017
- GitHub url of notebook - https://github.com/taaha/SpaceX-Capstone-Project/blob/master/EDA%20_using_SQL.ipynb

Build an Interactive Map with Folium

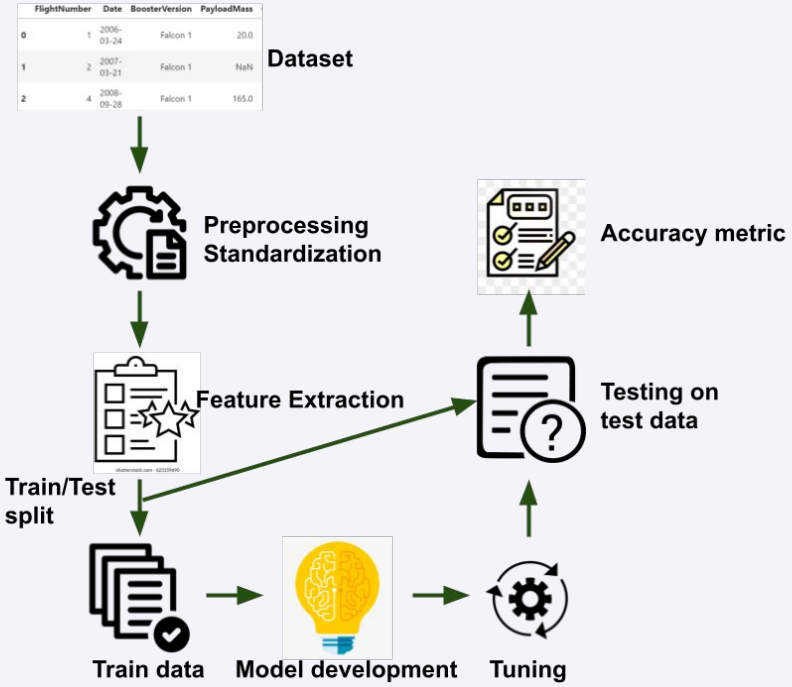
- Circles and markers were used to show location of launch sites on map
- Marker cluster was used to show number of launches on each site. Also markers were added to marker cluster which on zoom show different launches in colors showing if launch was successful (color=green) or failure (color=red)
- Distance of one launch site to the coast was also shown. Polyline was placed between site and coast
- GitHub url of notebook - https://github.com/taaha/SpaceX-Capstone-Project/blob/master/Visual_Analytics_with_Folium.ipynb

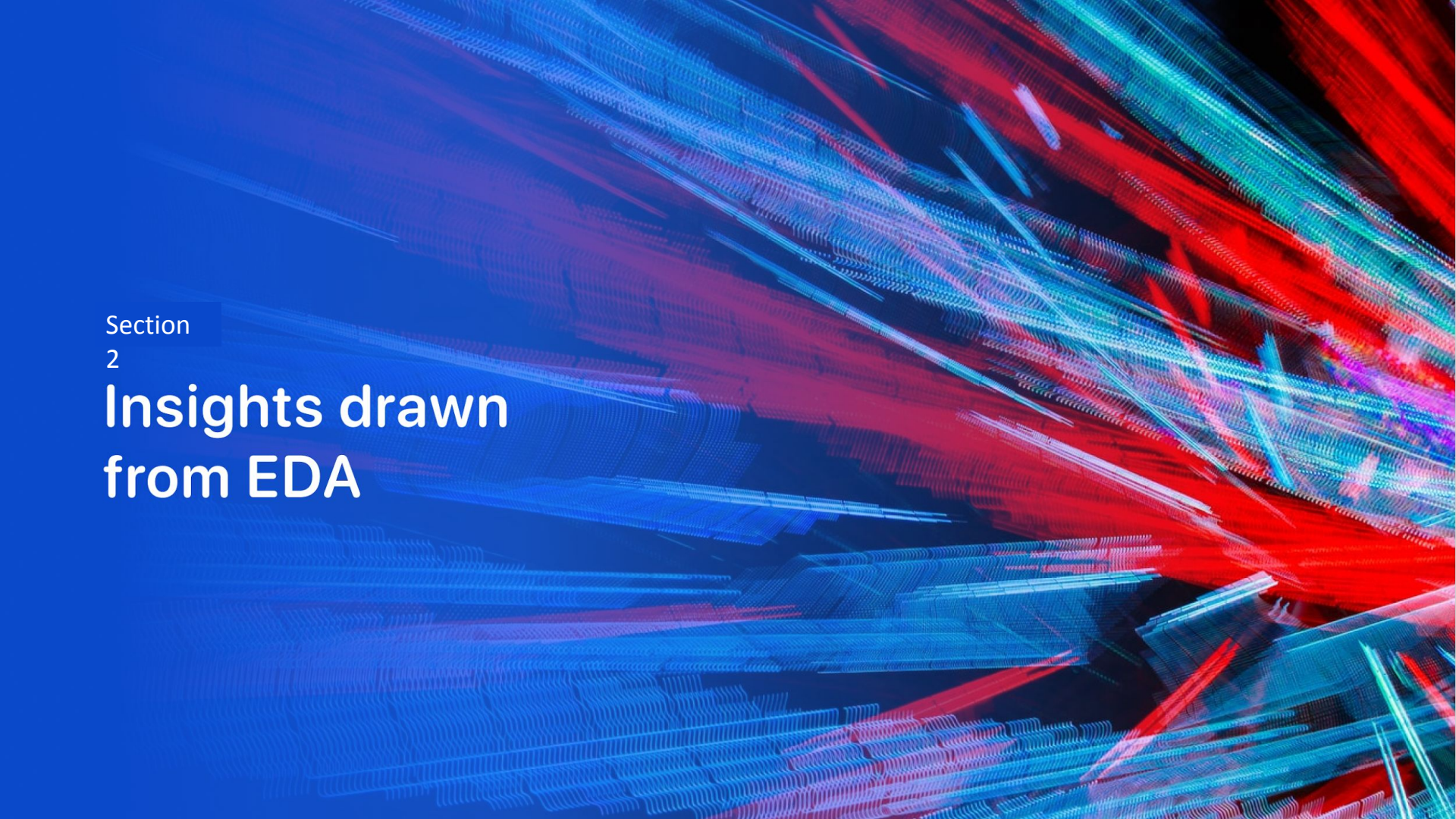
Build a Dashboard with Plotly Dash

- A dropdown was added to select all launch sites or any individual site
- Pie chart showed successful launches of all launch sites
- Piechart also showed success rate of individual site
- A slider was added to select payload range
- Scatter plot was used to show launch outcome for payload of different booster versions. Payload range could be changed by slider
- Github url of code - https://github.com/taaha/SpaceX-Capstone-Project/blob/master/spacex_dash_app.py
- Dataset url in appendix

Predictive Analysis (Classification)

- Logistic Regression, SVM, Decision Tree and KNN classifiers were built using scikit-learn library
- Dataset was standardized and splited into train and test data
- Classifier was fitted on training data
- Grid search CV was used on train data for tuning and to get the best value of parameters. 10 folds were used
- Model was tested with tuned parameters on test data and accuracy score and confusion matrix were computed
- All classifiers gived accuracy 0f 83.33%
- GitHub url of notebook - https://github.com/taaha/SpaceX-Capstone-Project/blob/master/Machine_Learning_Prediction_using_Scikit.ipynb

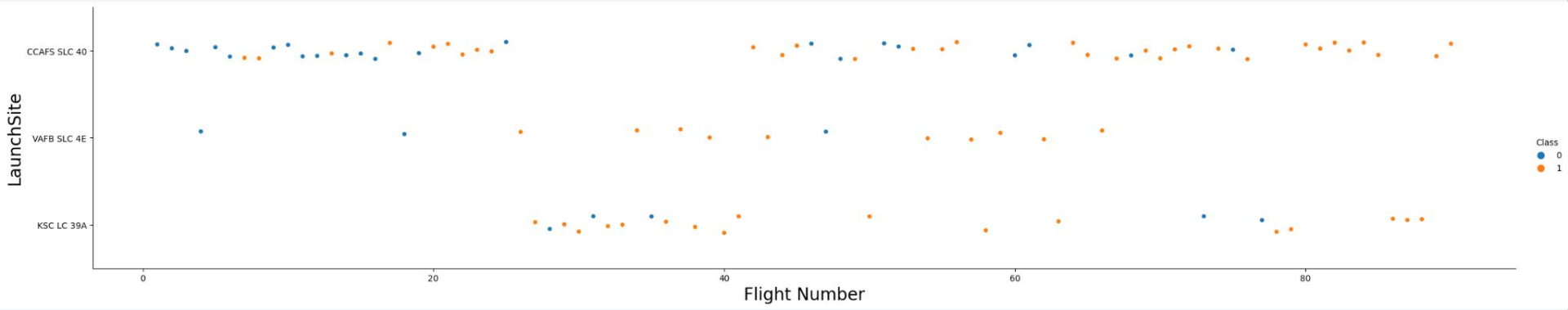




Section 2

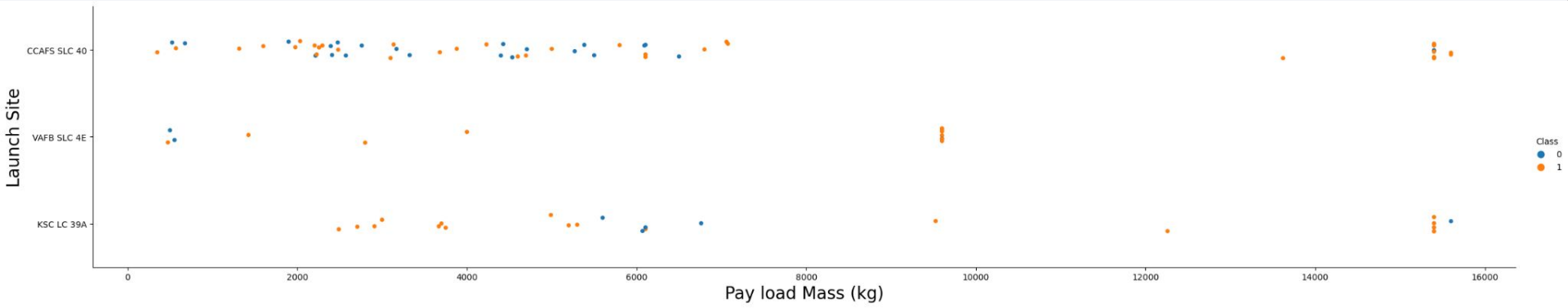
Insights drawn from EDA

Flight Number vs. Launch Site



- For sites CCAFS SLC 40 and VAFB SLC 4E, landing success rate increases with increase in flight number
- For KSC LC 39A there is no clear relationship between flight number and landing outcome. However, This launch site has good success rate

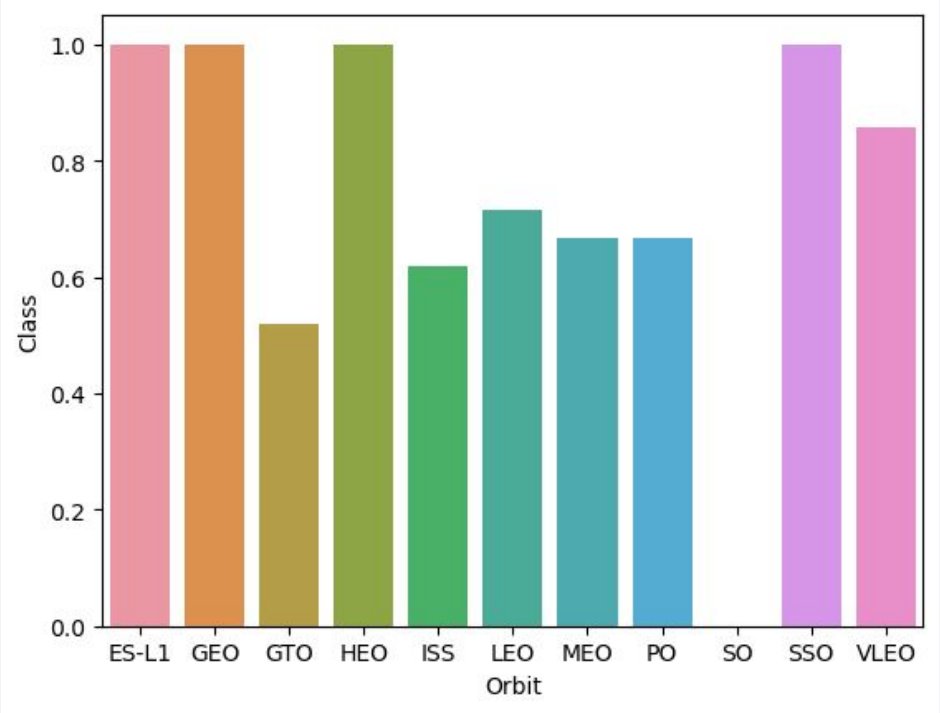
Payload vs. Launch Site



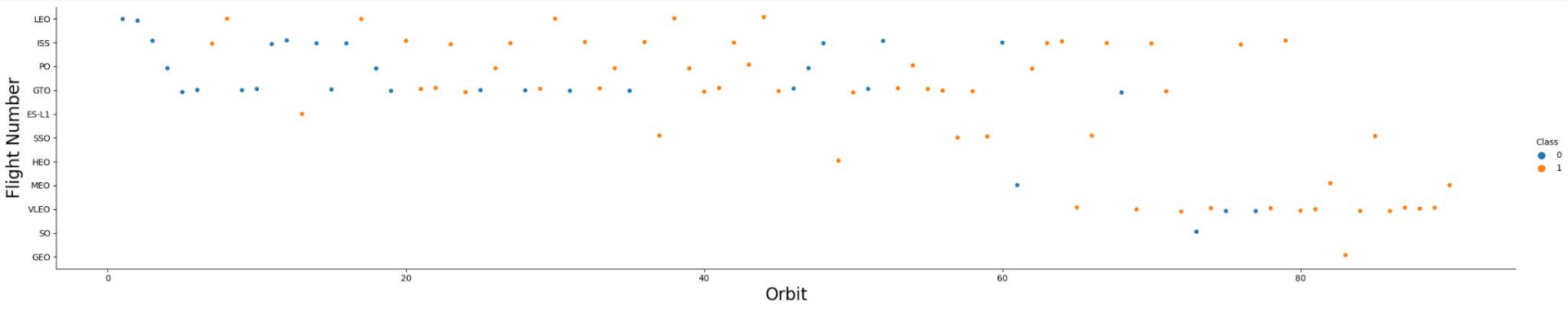
- VAFB SLC 4E, lauch site has no launched rockets with payload greater than 10000
- There is no clear relationship between flight number, payload and landing outcome.

Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO have 100% success rate
- GTO has the lowest success rate

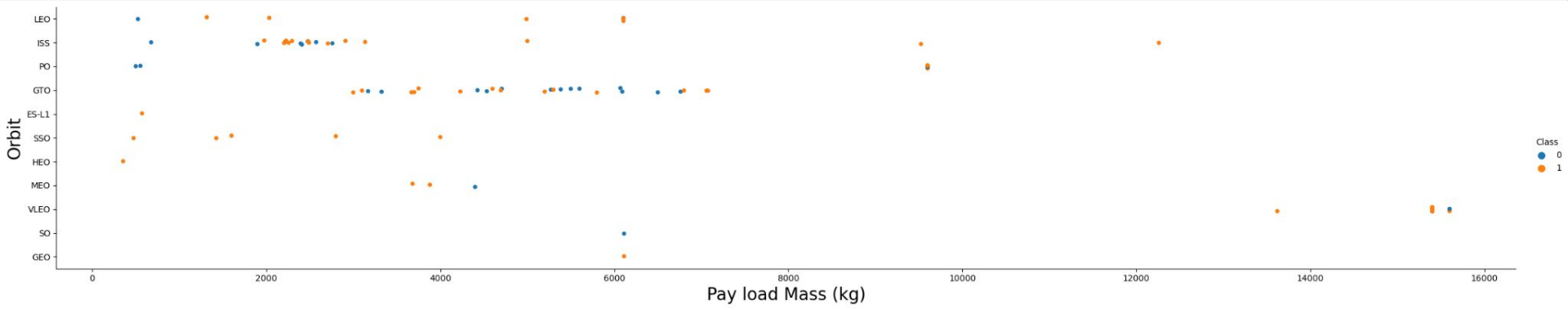


Flight Number vs. Orbit Type



- For LEO orbit, success rate increases with flight number
- For GTO orbit their is no clear relationship between success and fligh number
- For MEO, VLEO, SO and GEO orbits, SpaceX has started launches only recently.
- Initial space launches at start of company were mostly in LEO, ISS, PO and GTO orbits.

Payload vs. Orbit Type

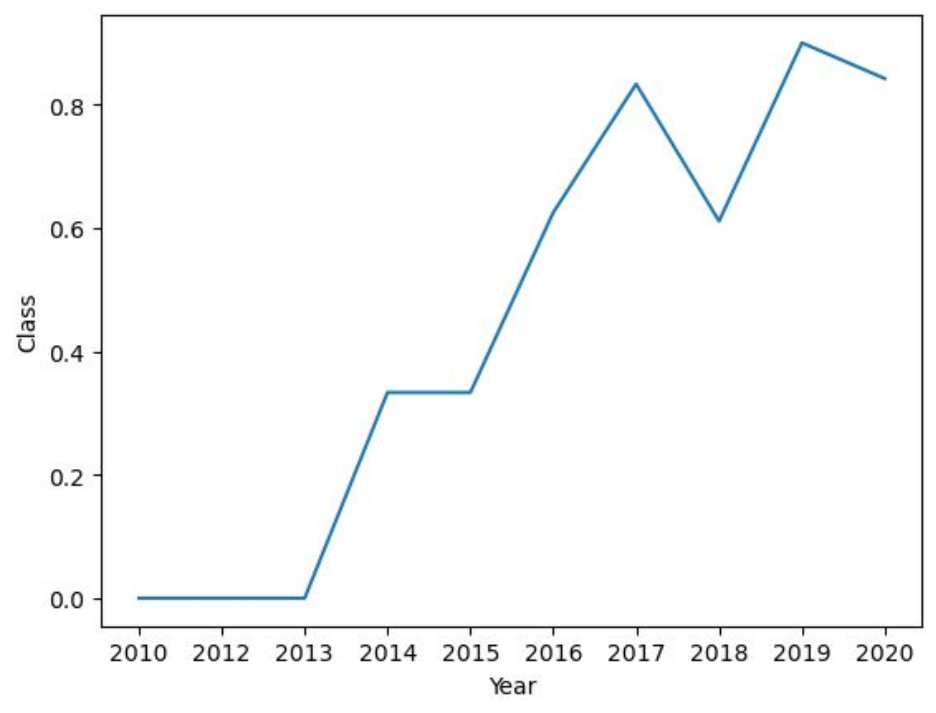


- For heavy payloads, PO, ISS and LEO orbits have good success rate
- For GTO orbit their is no clear realtionship between payload and success rate
- Payload greater than 8000 kg are only launched in ISS, PO and VLEO orbits

Launch Success Yearly Trend

- Average yearly success rate has roughly increased with time from 2013 to 2020

- There was a dip in yearly success rate in 2018 due to high number of failed launches



All Launch Site Names

- DISTINCT function is used to get distinct names of launch sites

```
%%sql
select distinct "Launch_Site" from spacextbl;
```

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

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

Launch Site Records Begin with 'CCA'

```
%%sql
select * from spacextbl
where "Launch_Site" like "%CCA%" limit 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- WHERE clause is used to filter records
- LIKE operator with wildcards (%) is used to find string `CCA` in launch sites

Total Payload Mass

- WHERE clause is used with LIKE operator to filter only those records which contain ‘NASA (CRS)’ string in Customer column
- SUM function is used to get sum of entries in payload_mass__kg_ column

```
%%sql
select sum(payload_mass__kg_) from spacextbl
where "Customer" like "%NASA (CRS)%";
```

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

sum(payload_mass__kg_)
48213

Average Payload Mass by F9 v1.1

- WHERE clause is used with LIKE operator to filter only those records which contain ‘F9 v1.1’ string in Booster_Version column
- AVG function is used to get average of entries in payload_mass__kg_ column

```
%%sql
select avg(payload_mass__kg_) from spacextbl
where "Booster_Version" like "%F9 v1.1%";
```

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

avg(payload_mass__kg_)
2534.6666666666665

First Successful Ground Landing Date

- WHERE clause is used with LIKE operator to filter only those records which contain ‘Success’ string in Mission_Outcome column and ‘ground’ string in Landing_Outcome column
- Two conditions are linked by AND operators
- LIMIT function is used to get first record (i.e lowest date)
- Alternatively MIN function can also be used

```
%%sql
select "Date" from spacextbl
where "Mission_Outcome" like "Success"
and "Landing _Outcome" like "%ground%"
limit 1;
```

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

Date
22-12-2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- WHERE clause is used with LIKE operator to filter only those records which contain ‘Success (drone’ string in Landing_Outcome column
- BETWEEN operator is used to select records with payload mass in range 4000 to 6000
- Two conditions are linked by AND operators

```
%%sql
select "Booster_Version" from spacextbl
where "Landing _Outcome" like "%Success (drone%"
and payload_mass__kg_ between 4000 and 6000;
```

```
 * 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

- GROUP BY clause is used to arrange data into groups having same Mission_Outcomes
- COUNT function returns number of rows for each value of Mission_Outcome

```
%%sql
select "Mission_Outcome",count(*) from spacextbl
group by "Mission_Outcome";
```

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

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

Boosters Carried Maximum Payload

- Subquery is embedded in the WHERE clause to filter records in which payload mass is maximum
- MAX function is used to get the maximum payload mass in the payload_mass__kg_

```
%%sql
select "Booster_Version" from spacextbl
where payload_mass__kg_ == (select max(payload_mass__kg_) from spacextbl);
```

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

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

2015 Launch Records

```
%%sql
select (substr(Date,4,2)) as month_number, "Landing _Outcome", "Booster_Version", "Launch_Site" from spacextbl
where substr(Date,7,4)='2015' and "Landing _Outcome" like "%Failure (drone%";
```

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

month_number	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

- substr() is used to extract part of string from the Date string to get month and year
- WHERE clause is used with AND operator to filter records having ‘Failure (drone’ string in Landing_Outcome column and having year 2015 in Date string

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

- GROUP BY was used with HAVING clause to filter records within a date range using BETWEEN function

- ORDER BY function was used with DESC to arrange results in descending order

```
%%sql
select "Landing _Outcome",count(*) as count from spacextbl
group by "Landing _Outcome"
having "Date" between '04-06-2010' and '20-03-2017'
order by count desc;
```

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

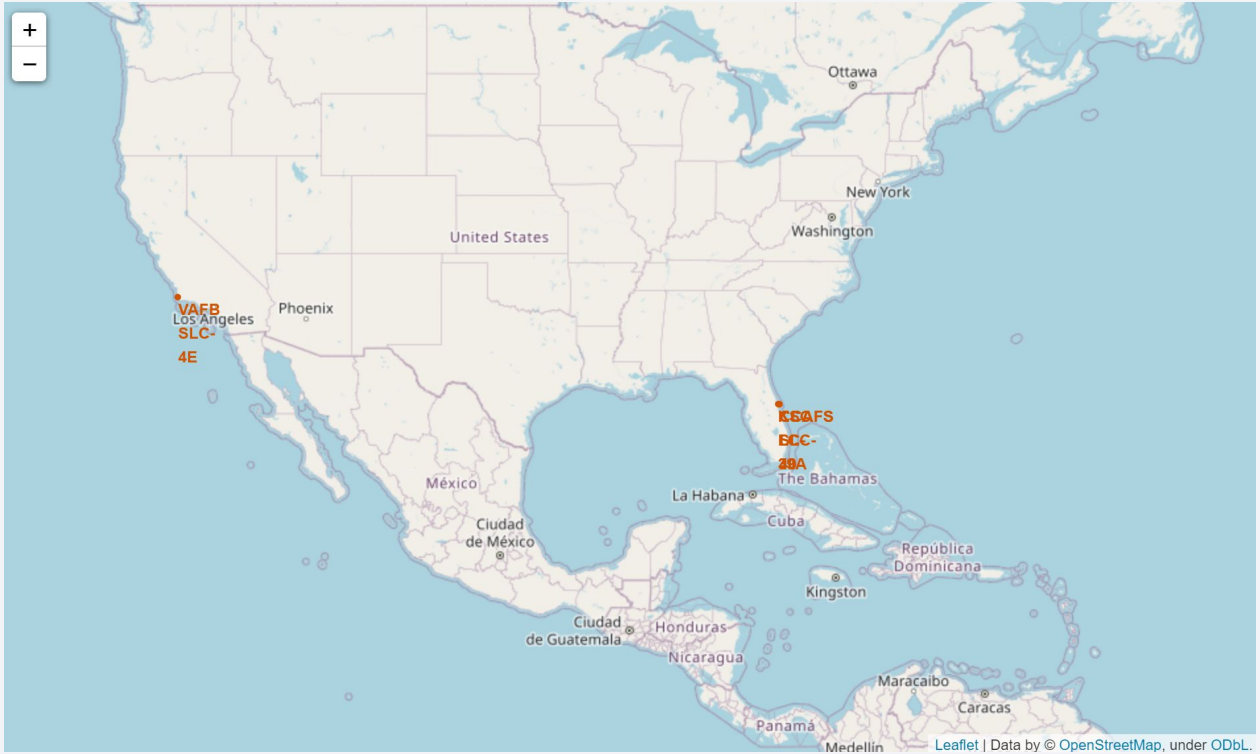
Landing _Outcome	count
Success (drone ship)	14
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Failure (parachute)	2
No attempt	1

Section 3

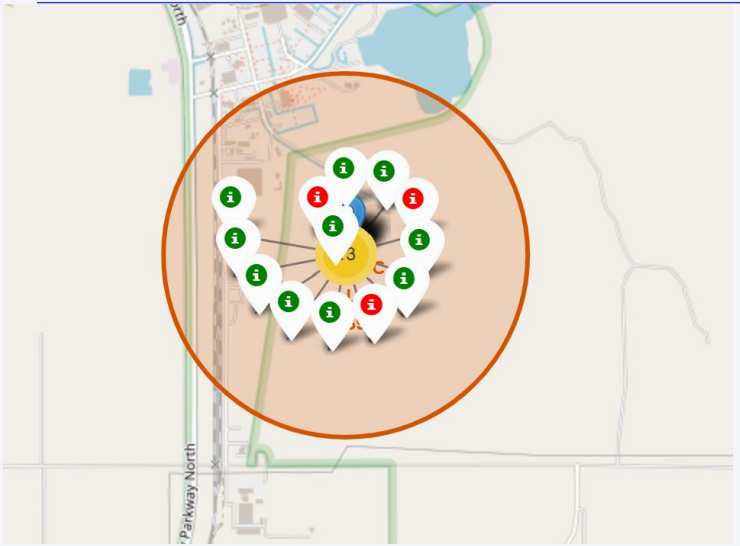
Launch Sites Proximities Analysis

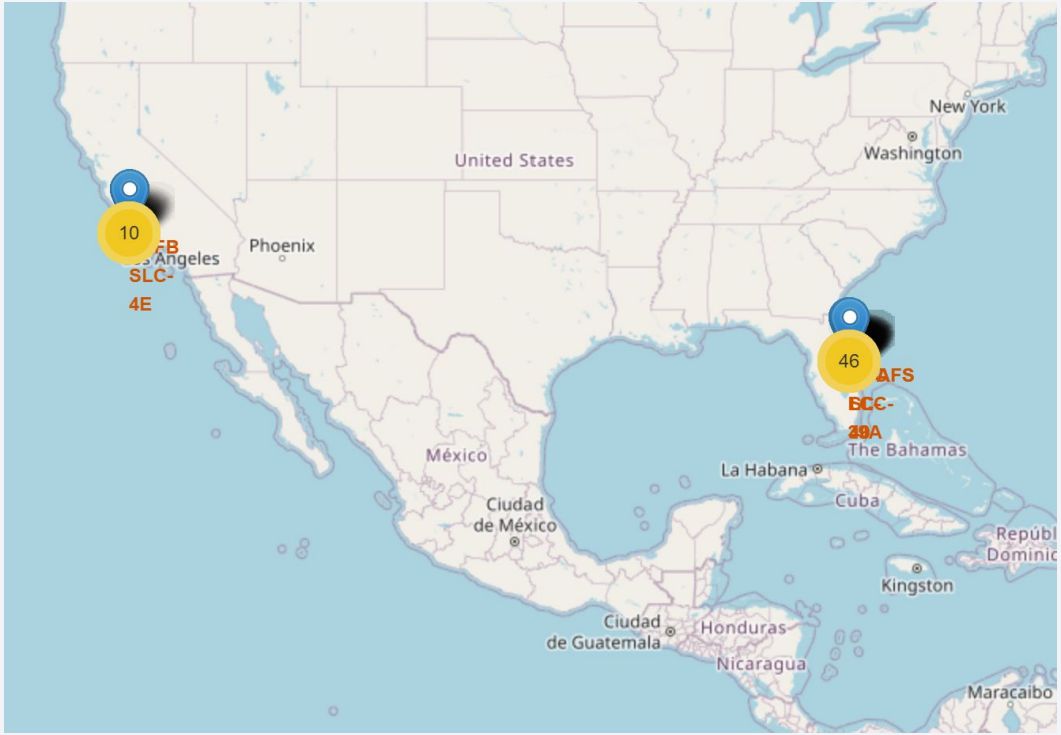
Location of all Launch Sites

- Launch sites are in coastal area as landing may occur in sea
- Launch sites are also close to equator
- Three launch sites are on the eastern coast and one is on the west coast



Successful and failed launches on each site

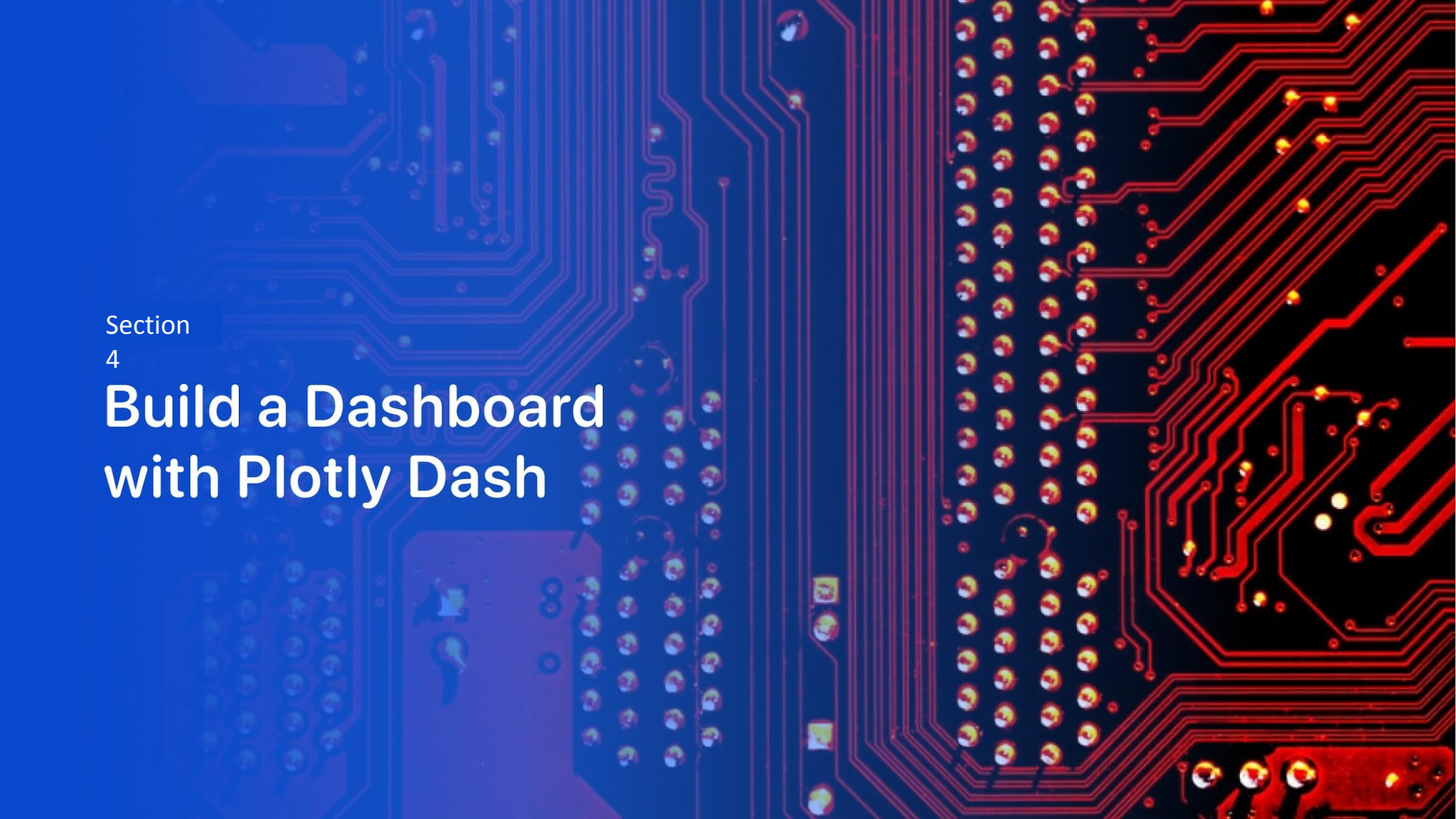




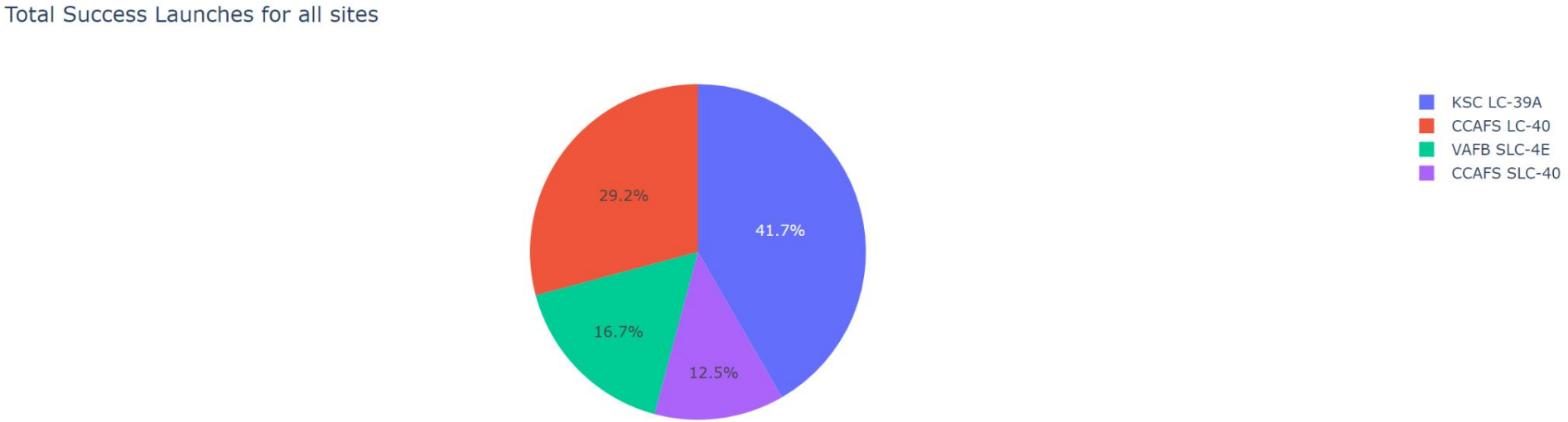
- We can observe success rate of each site e.g VAFB SLC-4E has poor success rate

CCAFS SLC-40 proximity to coast

- It can be seen that CCAFS SLC-40 distance to coast is approximately 0.9 km
- Similarly other launch sites are located close to the coast

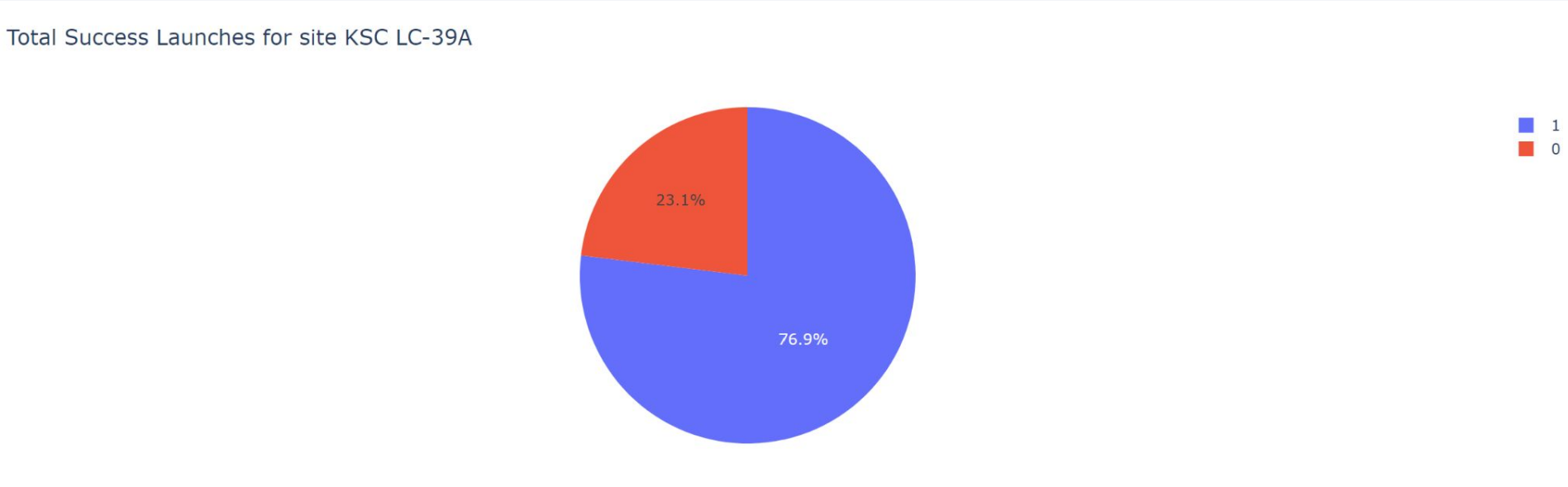

Section
4
Build a Dashboard
with Plotly Dash

Successful launches for all sites



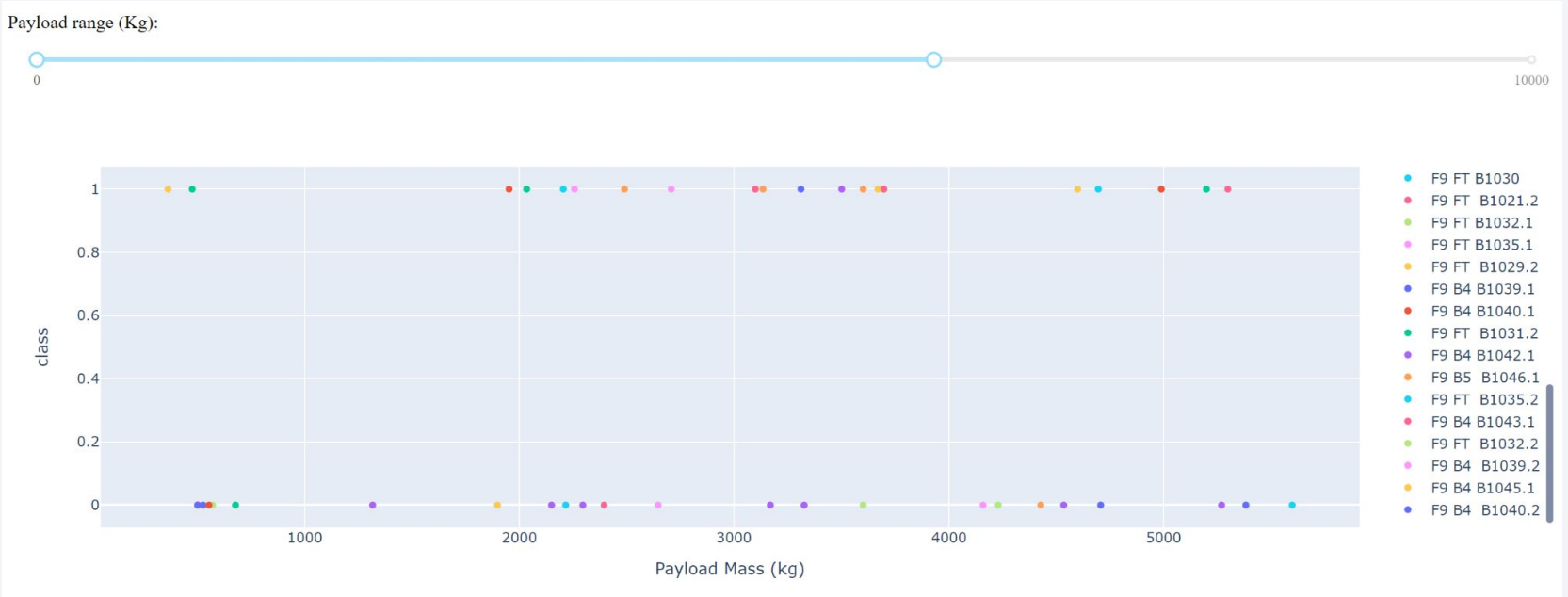
- KSC LC-39A has highest number of successful launches while CCAFS SLC-40 has lowest number of successful launches

Launch Site with highest success rate



- KSC LC-39A has highest success rate with 76.9% successful launches

Payload vs Launch Outcome



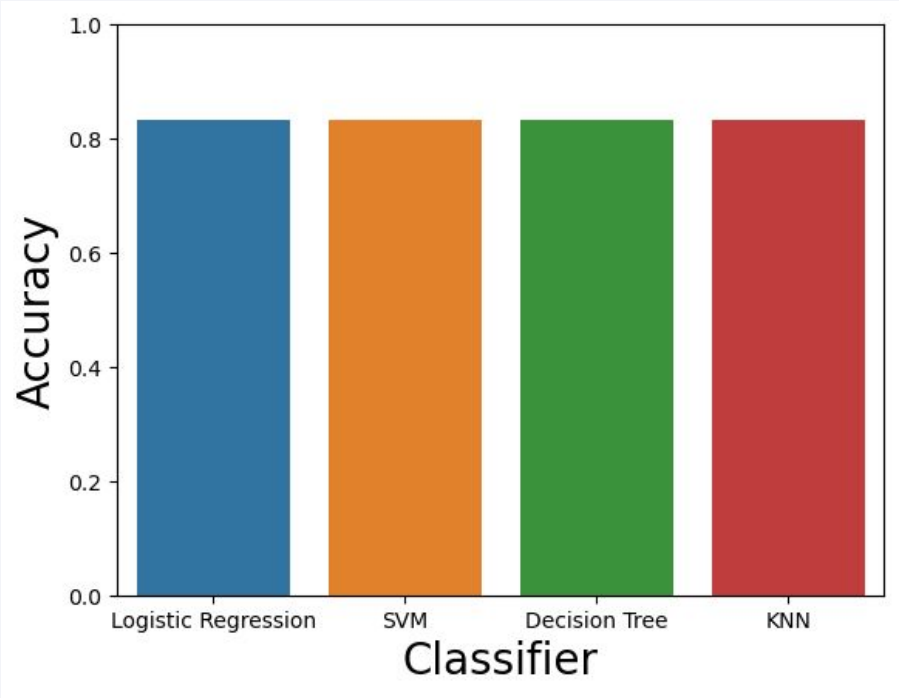
- For different payloads, it was observed that payload less than 6000 kg have high success rate as compared to payloads above 6000 kg. Payload range can be adjusted using slider.

Section
5

Predictive Analysis (Classification)

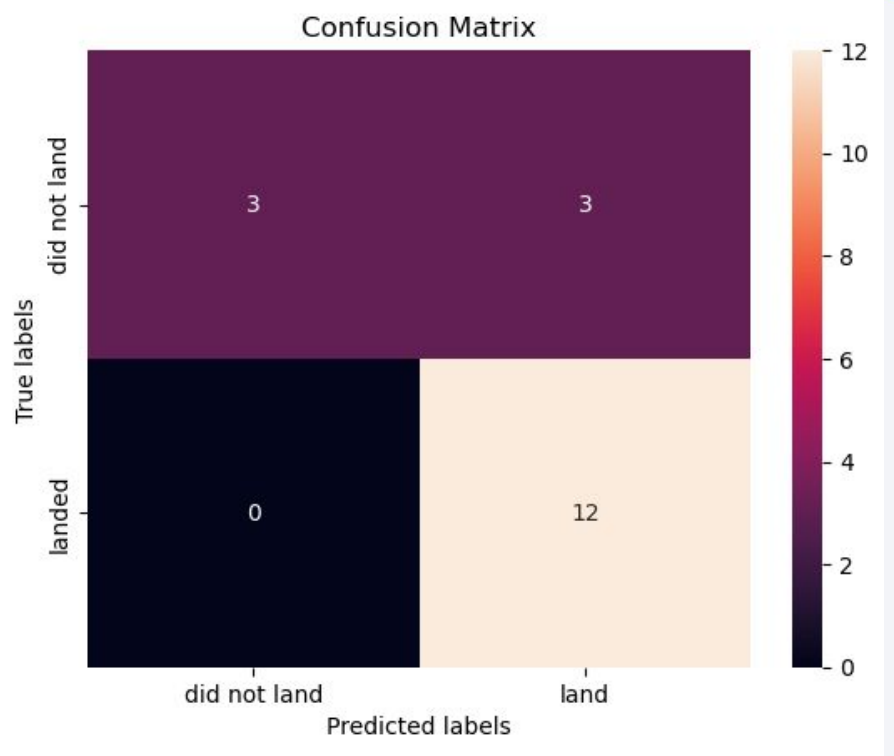
Classification Accuracy

- Logistic regression, SVM, Decision Tree and KNN algorithms were used
- All classifiers are showing the same accuracy of 83.33 %



Confusion Matrix

- All classifiers give the same confusion matrix.



Conclusions

- All classifiers have the same accuracy score of 83.33 %. This means that we can predict with 83.33% accuracy if the rocket will land or not.
- Inaccuracy of classifier is due to its tendency to give False Positives (FP)
- Classifier does not give any False Negatives (FN) and is highly accurate when True label is landed.

Appendix

- Dataset used for plotly dashboard - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_dash.csv
- Github link of project files - https://github.com/taaha/SpaceX-Capstone-Project
- Follow me on Github - https://github.com/taaha

Thank you!