DATA SCIENCE CAPSTONE PROJECT

Raghuvir Jonnagiri

Sep 05 2021

OUTLINE

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

EXECUTIVE SUMMARY

Methodologies:

- Launch data of spacex was collected using api
- Missing data was replaced by mean values and categorical values were converted to numerical values. Variables like orbit and LaunchSite were converted to one hot encoded variables for data analysis.
- SQL and Folium maps visualization were used for exploratory data analysis
- Four ML models of SVM, Decision Tree, KNN neighbors classification and Logistic Regression were used to predict the mission outcome of a launch. GridSearch was used to find the best parameters.

Summary of all results:

- Mission success rate has been steadily increasing since 2013.
- Launch sites are all close to coast and have rail/road close to the site. All the launch sites are at least 10 km away from closest cities
- In general, missions with payloads of more than 8000 kg have high success rate
- SSO orbit missions have highest success rate where as GTO orbit missions have least success rates
- All four models of ML have same accuracy on this data set of about 83%.

INTRODUCTION

Project background and context :

- Space X launches payloads of different weights at different orbits from different launch sites. Their main advantage over the competition is their ability to reuse the boosters and reduce the overall cost. This project is to study the uncertainty in being able to land the booster successfully for reuse so that spacex can maintain their competitive advantage.

Problems you want to find answers :

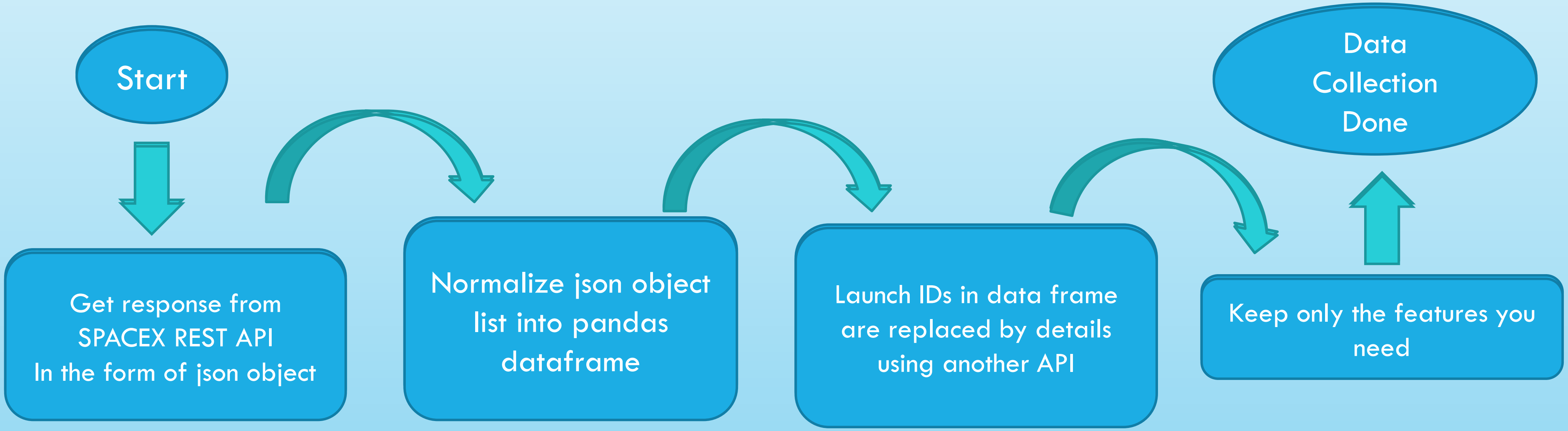
- Can we predict the success of a booster being able to be recovered?
- How does this mission success vary with payload and orbit type?
- Is the SpaceX program improving over the time?

METHODOLOGY

- Data collection
- Data wrangling
- Exploratory data analysis (EDA) using visualization and SQL ‘’
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analysis using classification models

DATA COLLECTION

Data collection – SpaceX API



Start

Get response from SPACEX REST API In the form of json object

Normalize json object list into pandas dataframe

Launch IDs in data frame are replaced by details using another API

Keep only the features you need

Data Collection Done

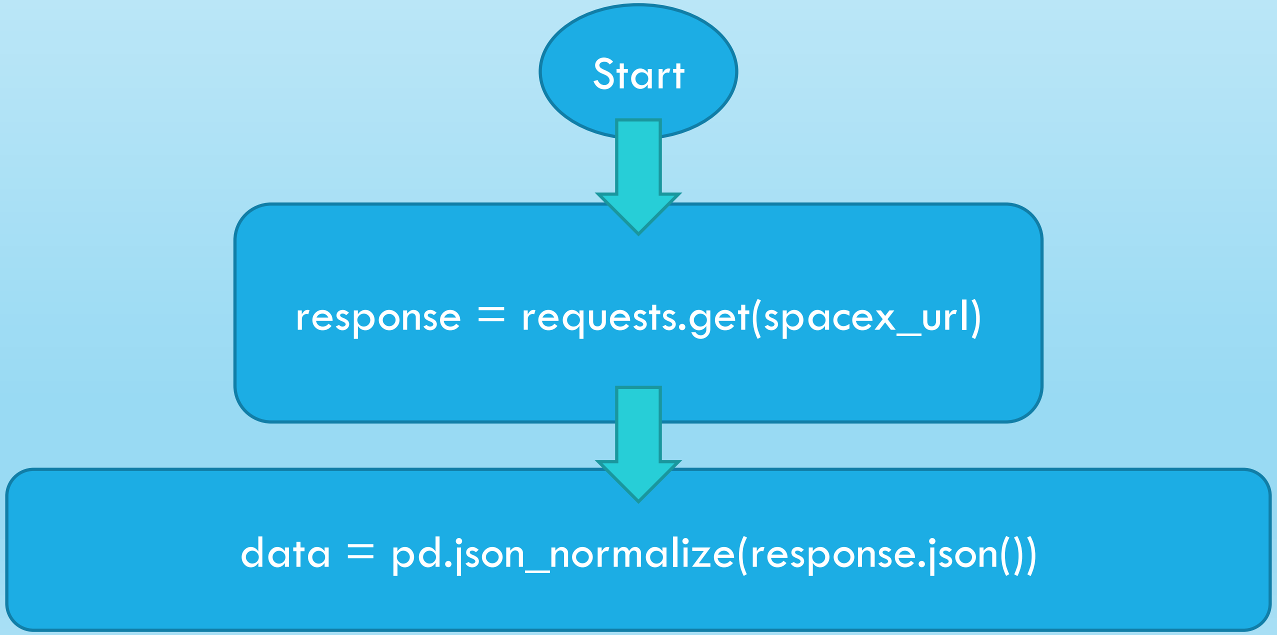
- Code URL :
- https://github.com/ravirejo/DataScienceIBM_Course/blob/master/SpaceXDataCollection.ipynb

DATA COLLECTION – SPACEX API

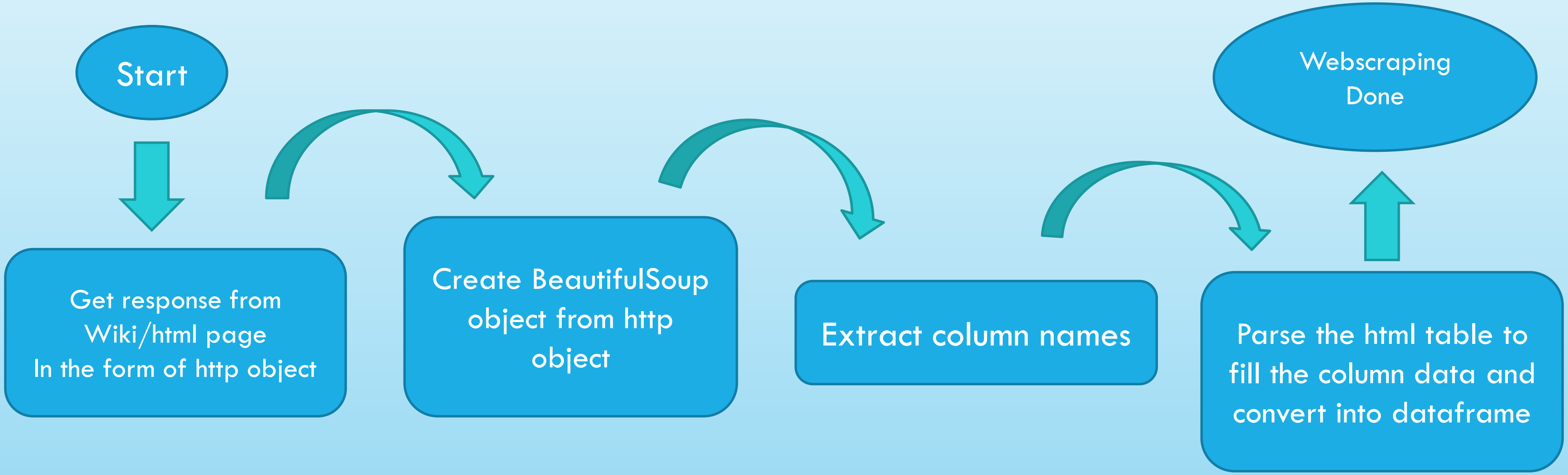
SpaceX REST API endpoint = api.spacexdata.com/v4/launches/past

Code URL :

https://github.com/ravirejo/DataScienceIBM_Course/blob/master/SpaceXDataCollection.ipynb


Start
response = requests.get(spacex_url)
data = pd.json_normalize(response.json())

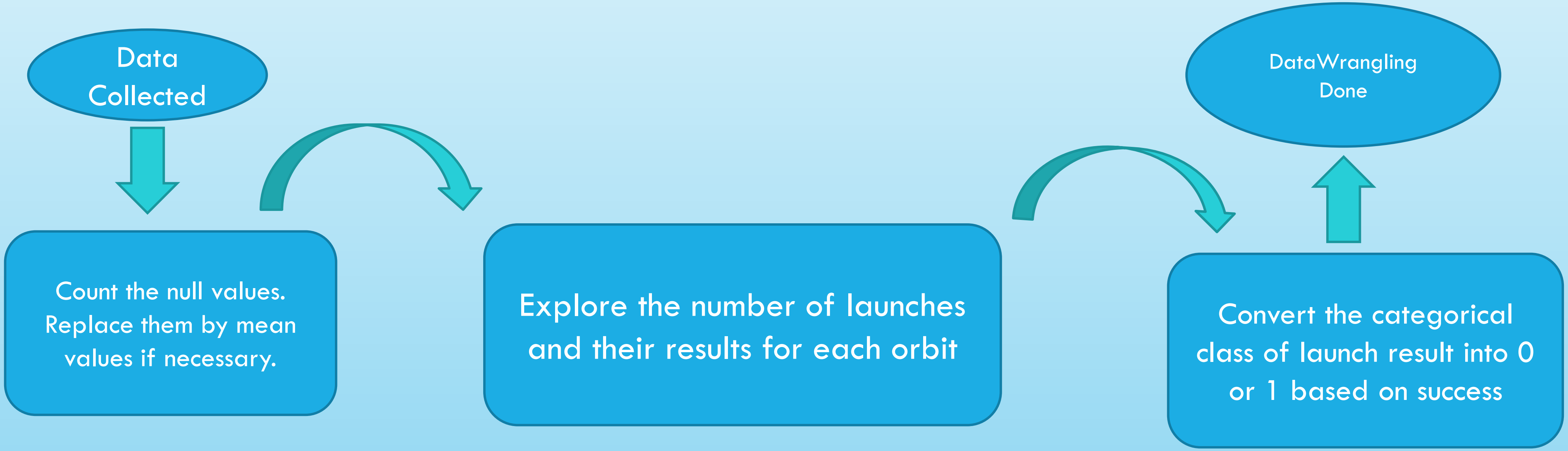
DATA COLLECTION – WEB SCRAPING


Start
Get response from Wiki/html page In the form of http object
Create BeautifulSoup object from http object
Extract column names
Parse the html table to fill the column data and convert into dataframe
Webscraping Done

Code URL :

https://github.com/ravirejo/DataScienceIBM_Course/blob/master/SpaceX_Webscraping.ipynb

DATA WRANGLING


Data Collected

Count the null values. Replace them by mean values if necessary.

Explore the number of launches and their results for each orbit

Convert the categorical class of launch result into 0 or 1 based on success

DataWrangling Done

Code URL :

https://github.com/ravirejo/DataScienceIBM_Course/blob/master/SpaceXDataWrangling.ipynb

EDA WITH DATA VISUALIZATION

Payload mass was scatter plotted against Flight number with success as color to study the progression of payload mass with flight number and their success.

Launch Site was scatter plotted against Flight number with success as color to study the success of missions launched at each site as the flight number increases.

Launch Site was scatter plotted against Payload mass to see if a specific launch site was associated with large or small payloads.

Bar chart was plotted of average mission success for each orbit to study each orbit's success rate

Orbit type was scatter plotted against Flight number to see if there were any preferred orbits for earlier and later flights.

Orbit type was also scatter plotted against Payload mass to see if there were any preferred payload mass for each orbit.

Finally, mission success is plotted against year to see if the missions have improving success rate with time.

Code URL :

https://github.com/ravirejo/DataScienceIBM_Course/blob/master/SpaceXVisualization.ipynb

EDA WITH SQL

Names of unique launch sites were queried using *distinct launch_site from SPACEXTBL*

Five launch sites with names starting with ‘CCA’ were obtained using *from SPACEXTBL where launch_site like 'CCA%’ limit 5*

Total payload mass launched by a certain site(NASA CRS) was obtained using *sum(payload_mass__kg_) from SPACEXTBL where customer like '%CRS%’*

Average payload mass launched by a certain booster version (F9 v1.1)was obtained using *avg(payload_mass__kg_) from SPACEXTBL where booster_version like '%F9 v1.1%’*

First date of success landing in ground pad was obtained using *MIN(DATE) from SPACEXTBL where landing__outcome like '%Success%' and landing__outcome like '%ground%’*

Names of boosters successfully landed on drone ships with certain mass range (4000-6000) was obtained using *booster_version from SPACEXTBL where landing__outcome like '%Success%' and landing__outcome like '%drone%' and payload_mass__kg_ > 4000 and payload_mass__kg_< 6000*

Total successful and failed missions were obtained using *select mission_outcome, count(*) from SPACEXTBL group by mission_outcome*

All the booster versions which carry maximum payloads were obtained using *select booster_version from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)*

All the failed drone ship missions with their booster versions and launch sites in year 2015 were obtained using *select landing__outcome, booster_version, launch_site from SPACEXTBL where year(DATE) = 2015 and landing__outcome like '%Fail%' and landing__outcome like '%drone%’*

Landing outcomes were numbered between certain dates (June 04 2010 and March 20 2017) using *select landing__outcome, count(*) from SPACEXTBL where DATE > '2010-06-04' and DATE < '2017-03-20’ group by landing__outcome order by count(landing__outcome) desc*

Code URL :

https://github.com/ravirejo/DataScienceIBM_Course/blob/master/SpaceXSQL.ipynb

BUILD AN INTERACTIVE MAP WITH FOLIUM

All the launch sites were circle marked on the map using latitude and longtitude to get an idea of where the locations are and how spread out they are.

Each launch with color based on success was also marked. Since there were so many launches at the same site, marker cluster was used.

Polylines were drawn between launch sites and their nearest city/highway/shippinglane/railway.

Distances were calculated between the above mentioned locations and displayed on the polylines to see how far/close they are.

Code URL :
https://github.com/ravirejo/DataScienceIBM_Course/blob/master/SpaceXDashboard.ipynb

BUILD A DASHBOARD WITH PLOTLY DASH

Launch site dropdown menu was added to select a specific site and see the relevant pie chart or scatter plot.

Pie chart was plotted with default display of percentage of each launch sites in all the successful missions. If a specific launch site was selected, only success/failure percentage of that specific launch site was displayed on the pie chart.

Payload mass slider was added to focus only on the range of payload we are interested in and see how the trends change with different selection of payloads.

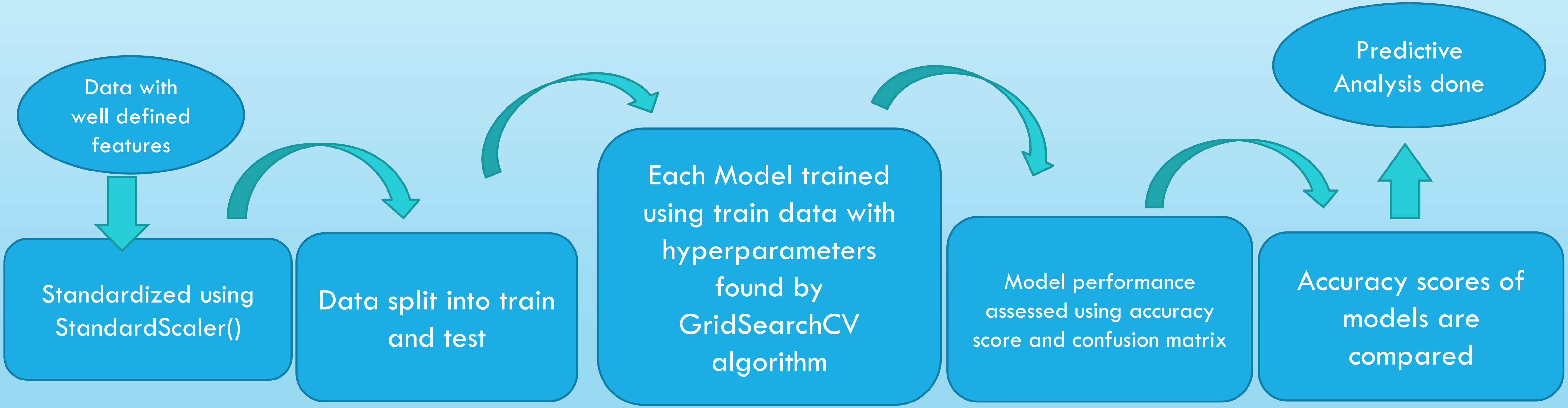
Finally, a scatter plot between payload mass and mission outcome class was displayed for each type of booster type colored differently. Default payload range was 0 to 10,000 kg which can adjusted with slider above.

Code URL :

https://github.com/ravirejo/DataScienceIBM_Course/blob/master/spacex_dash_app.py

PREDICTIVE ANALYSIS (CLASSIFICATION)

- Four ML methods(Support Vector Machine, Decision Tree, Logistic Regression and K-NearestNeighbors) were used to predict the outcome of the launch.
- First, Data was normalized using StandardScaler.
- Then, Data was split into train and test. With the training data, GridSearchCV was used to find the best hyperparameters of each ML model .
- Finally, performance with test data was assessed using confusion matrix.
- This accuracy score was compared for all four ML models.



Data with well defined features

Standardized using StandardScaler()

Data split into train and test

Each Model trained using train data with hyperparameters found by GridSearchCV algorithm

Model performance assessed using accuracy score and confusion matrix

Accuracy scores of models are compared

Predictive Analysis done

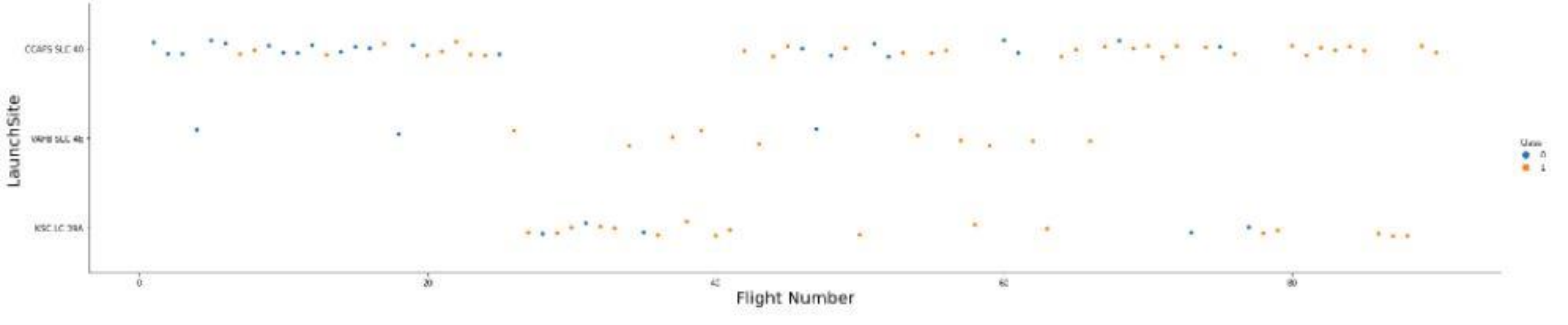
Code URL :
https://github.com/ravirejo/DataScienceIBM_Course/blob/master/SpaceX_ML%20copy%202.ipynb

RESULTS

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

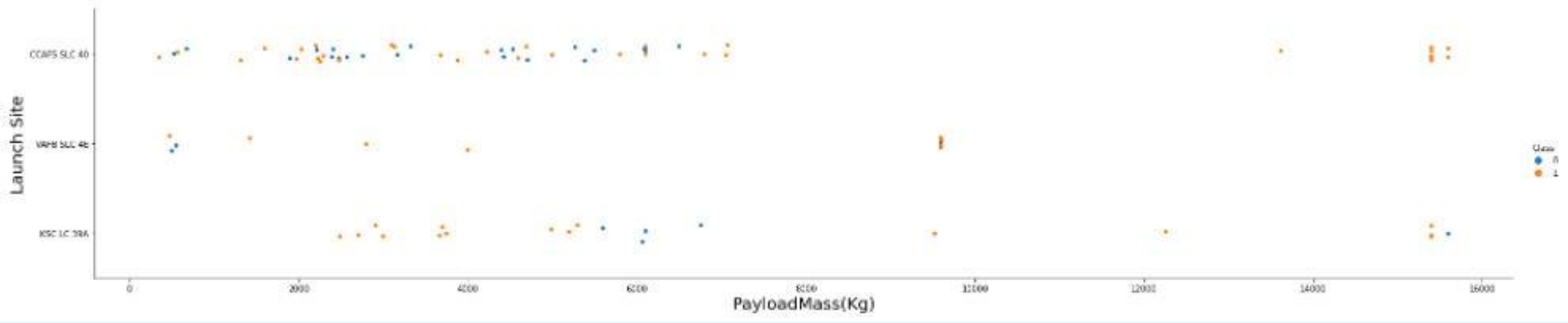
EDA WITH VISUALIZATION

FLIGHT NUMBER VS. LAUNCH SITE



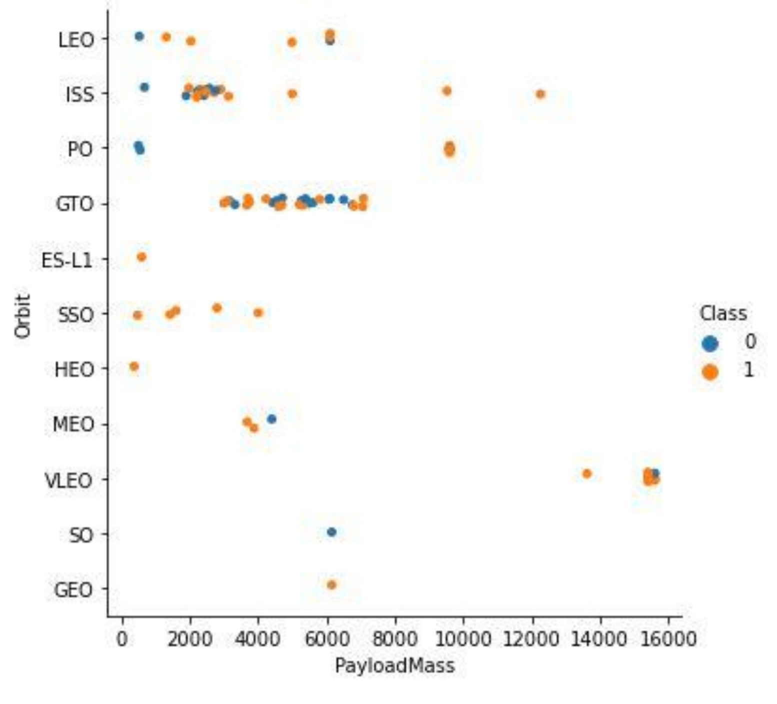
- ‘KSC 39A’ is relatively newer site
- All the launch sites have failures initially but success missions more later
- Launch Sites with most missions are in the order of ‘CCAPS SLC 40’, ‘KSC 39A’ and ‘VAFB SLC 4E’

PAYLOAD VS. LAUNCH SITE



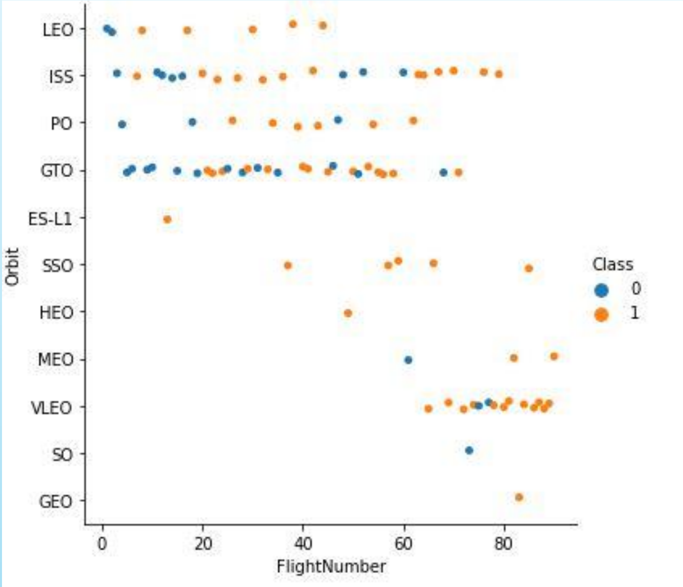
- Success rate is high for payloads higher than 8000 kg
- Most of the payloads seem to be either lower than 7000 kg or higher than 15000 kg
- Lot of payloads launched from ‘VAFB SLC 4E’ seem to be same around 9500kg

PAYLOAD VS. ORBIT TYPE



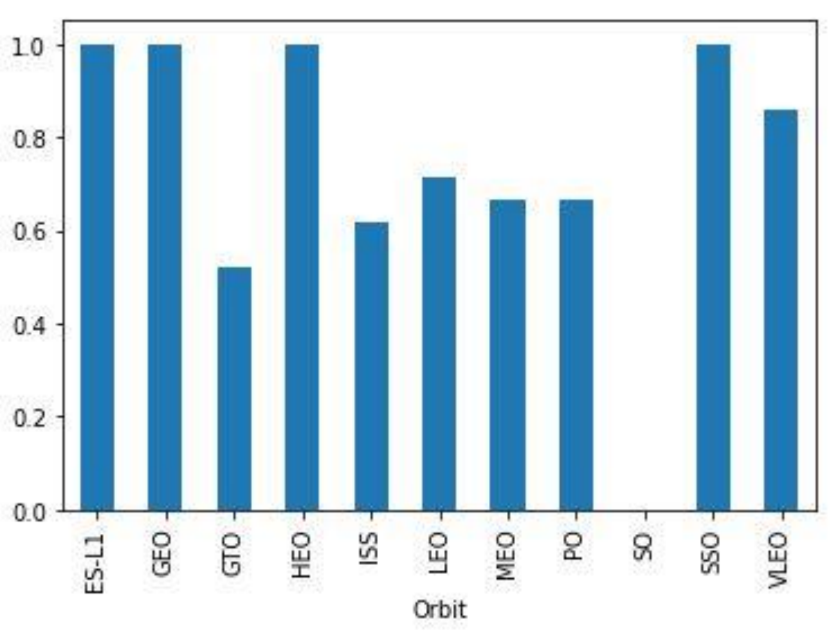
- SSO orbit seems to launch only low payload mass but with perfect mission record
- For GTO, success seems to be decreasing with payload mass
- For PO and ISS orbit, success increases with payload mass

FLIGHT NUMBER VS. ORBIT TYPE



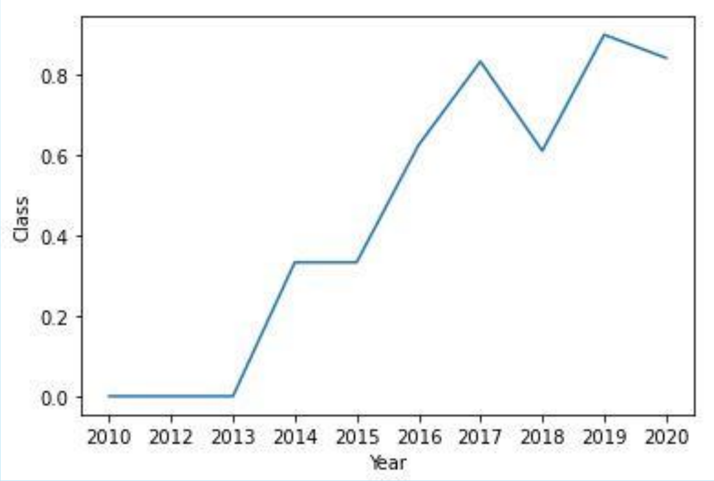
- For orbits like GTO, there is no clear relationship between GTO and number of flights
- For orbits like LEO, success rate increases with number of flights

SUCCESS RATE VS. ORBIT TYPE



- SSO, HEO, GEO, ES-L1 have perfect success rate
- GTO orbit has worst success record

LAUNCH SUCCESS YEARLY TREND



- Success rate has been increasing steadily since 2013
- Success rate in recent years has been 60-80%

EDA WITH SQL

ALL LAUNCH SITE NAMES

COMMAND:

```
select distinct launch_site from SPACEXTBL
```

RESULT:

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

EXPLANATION:

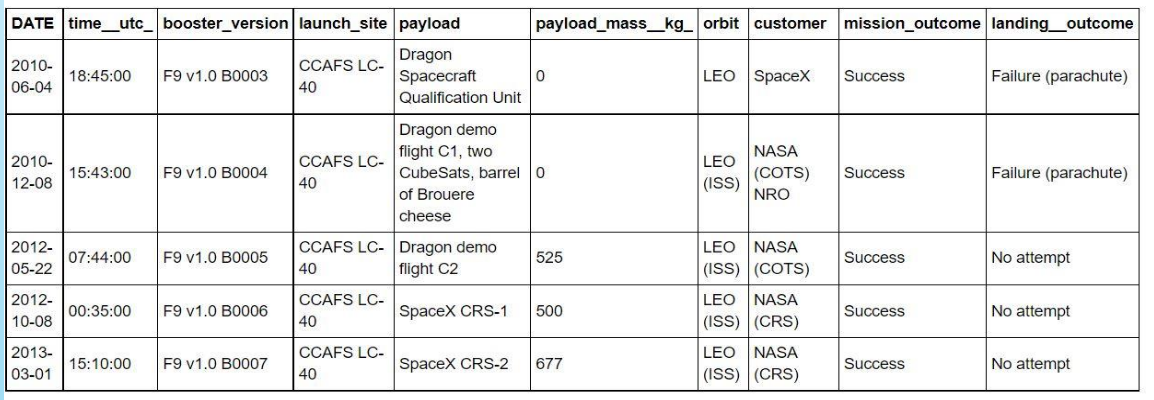
distinct is the key phrase for unique items. *launch_site* is the variable for launch site names. *SPACEXTBL* is the table name.

LAUNCH SITE NAMES BEGIN WITH `CCA`

COMMAND:

```
select * from SPACEXTBL where launch_site like 'CCA%‘ limit 5
```

RESULT:



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

EXPLANATION:

Where and *like* are the key phrases for specific word/string constraints .*limit* is to limit output.

TOTAL PAYLOAD MASS

COMMAND:

```
select sum(payload_mass__kg_) from SPACEXTBL where customer like '%CRS%'
```

RESULT:

48213

EXPLANATION:

sum is the key phrase for adding the variable *payload_mass__kg_*. *like* is used to limit output from only customers with CRS in their name (NASA CRS).

AVERAGE PAYLOAD MASS BY F9 V1.1

COMMAND:

```
select avg(payload_mass__kg_) from SPACEXTBL where booster_version like '%F9 v1.1%'
```

RESULT:

```
2534.666666
```

EXPLANATION:

Avg is the key phrase for averaging payload_mass__kg_ variable. *Where* and *like* are used to take only *booster_version* of *F9 V1.1* into consideration.

FIRST SUCCESSFUL GROUND LANDING DATE

COMMAND:

```
select MIN(DATE) from SPACEXTBL where landing__outcome like '%Success%' and
landing__outcome like '%ground%'
```

RESULT:

2015-12-22

EXPLANATION:

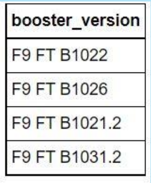
MIN(DATE) is the key phrase to find the date of first time of an event. *Where* and *like* are used to define the event where *landing_outcome* is a success and it is on *ground*.

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

COMMAND:

```
select booster_version from SPACEXTBL where landing__outcome like '%Success%' and
landing__outcome like '%drone%' and payload_mass__kg_ > 4000 and payload_mass__kg_<
6000
```

RESULT:



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

EXPLANATION:

Where and *like* are the key phrases used to define the event that *landing_outcome* should be a *success* and on a *drone*. <,> are used to constraint the *payload_mass* considered from *4000* to *6000.*

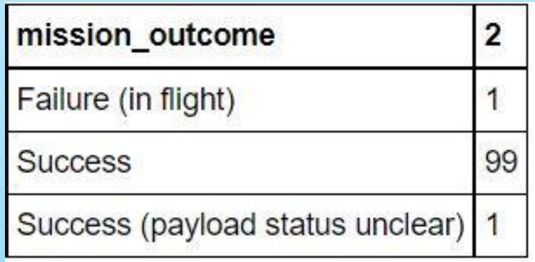
TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

COMMAND:

select mission_outcome, count(*) from SPACEXTBL

group by mission_outcome

RESULT:



mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

EXPLANATION:

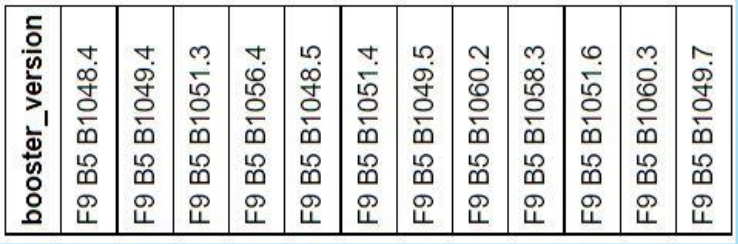
Count(*) is the key phrase to count all the items. *group_by* is the phrase to present the output in a grouped format of the variable *mission_outcome*

BOOSTERS CARRIED MAXIMUM PAYLOAD

COMMAND:

select booster_version from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)

RESULT:



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

EXPLANATION:

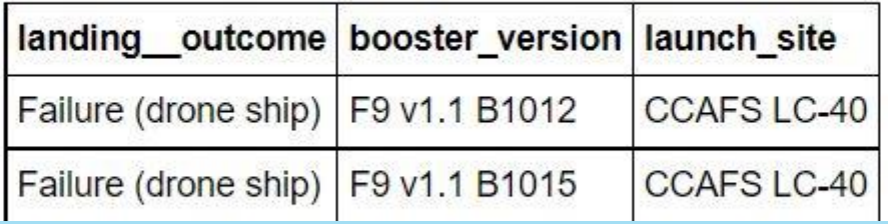
Max is the phrase to calculate the maximum value of *payload_mass_kg_*. *(select....)* is a subquery used to limit the *payload_mass_kg_* to only maximum values.

2015 LAUNCH RECORDS

COMMAND:

```
select landing__outcome, booster_version, launch_site from SPACEXTBL where year(DATE) =
2015 and landing__outcome like '%Fail%' and landing__outcome like '%drone%'
```

RESULT:



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

EXPLANATION:

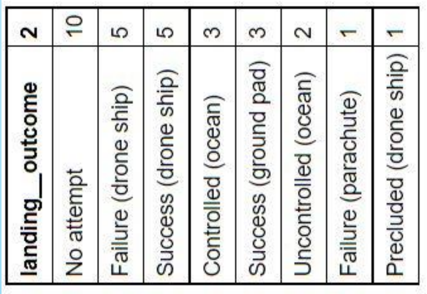
Year(DATE)=2015 is used to limit year of events considered only to 2015 where event represents a *landing_ outcome* which is a *fail* and is on a *drone.*

RANK SUCCESS COUNT BETWEEN 2010-06-04 AND 2017-03-20

COMMAND:

select landing__outcome, count(*) from SPACEXTBL where DATE > '2010-06-04' and DATE < '2017-03-20‘ group by landing__outcome order by count(landing__outcome) desc

RESULT:



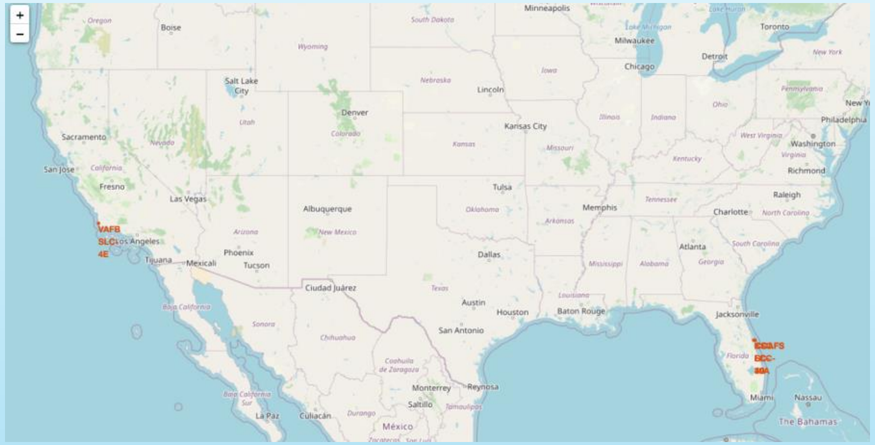
landing__outcome	2
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Uncontrolled (ocean)	2
Failure (parachute)	1
Precluded (drone ship)	1

EXPLANATION:

Order by is the key phrase to sort the output. *Desc* is used to have the order in descending. *DATE* is controlled by using <,>.

INTERACTIVE MAP WITH FOLIUM

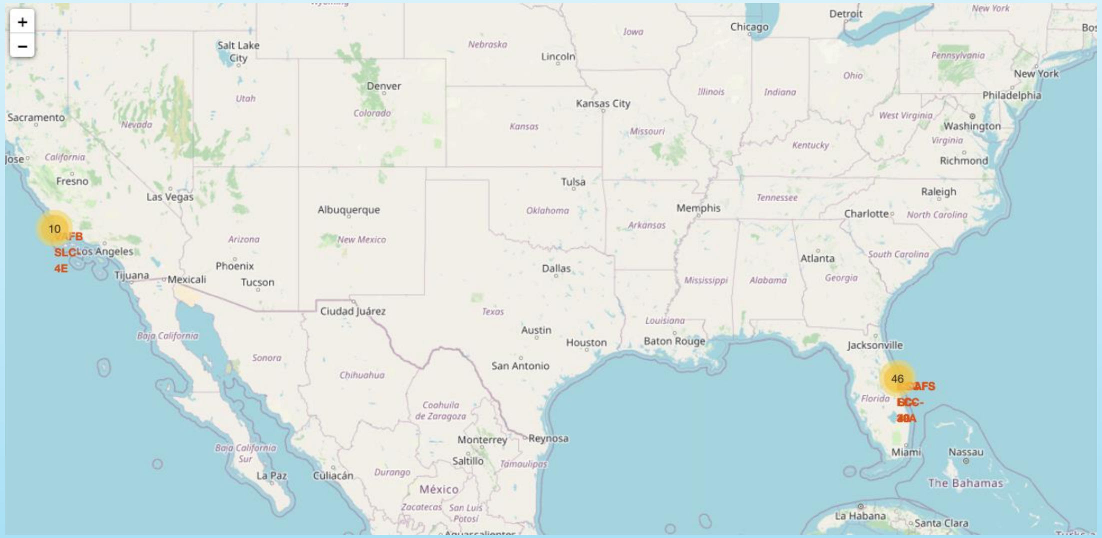
ALL LAUNCH SITES ON A GLOBAL MAP

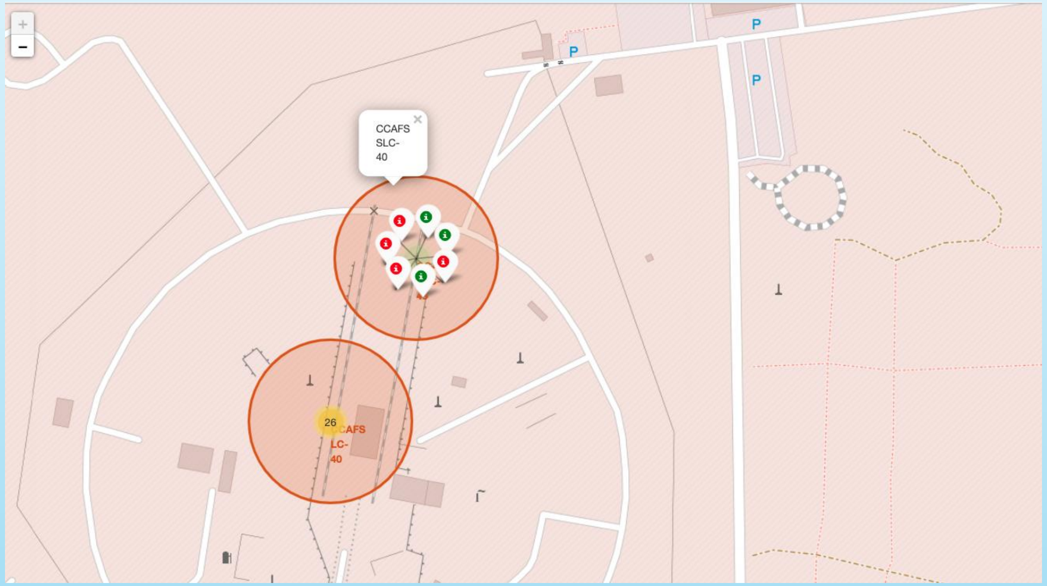


All the launch sites are near the equator. This is because earth surface speed is highest at the equator which can help the rocket travel faster after launch.

All the launch sites are near coastal region. This could be because of multiple reasons. One is to have the safety of water if a mission fails and crashes. Second is to be close to water transport for loads that are tough to carry by road/railways.

COLOR-LABELED LAUNCH RECORDS

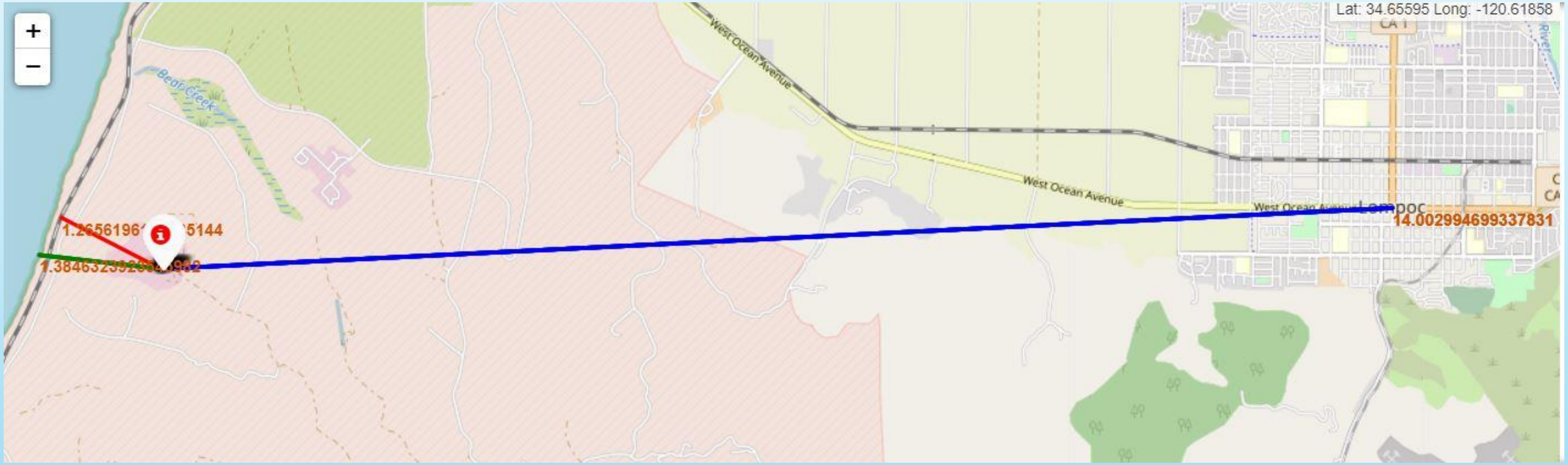




Marker cluster is used to show all the launch events on each launch site.

When Zoomed in, we can see each launch outcome as a success (green) or a failure (red).

LAUNCH SITE TO ITS TRANSPORT PROXIMITIES



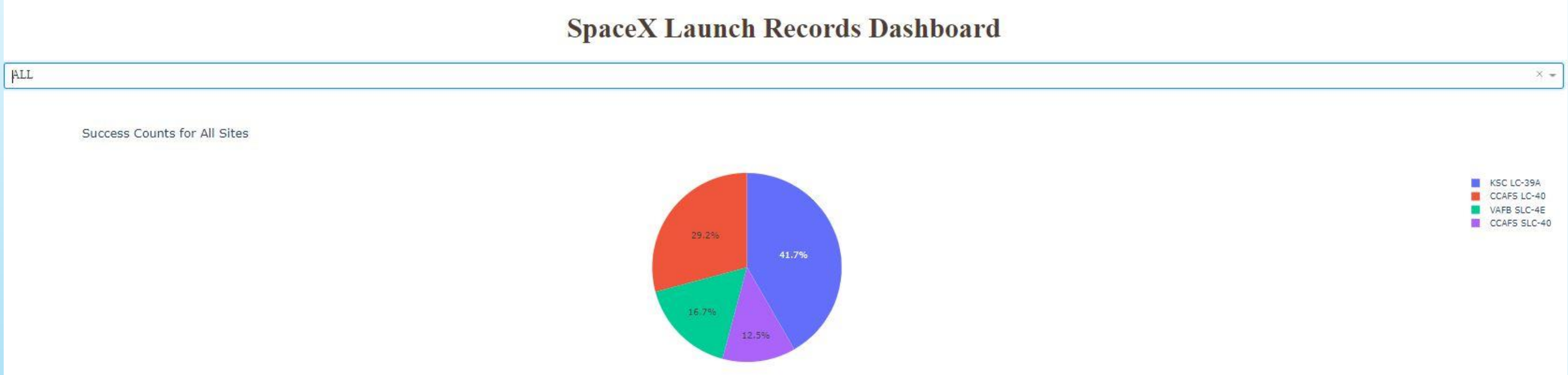
Polylines are used to depict the distances from nearest city/highway (blue), water (green) and railway (red).

Distances are also displayed in km.

Clearly,launch_site is intentionally very close to water/railways but also very far from nearest city to reduce the exposure of general population to launch mishaps.

BUILD A DASHBOARD WITH PLOTLY DASH

LAUNCH SUCCESS COUNT FOR ALL SITES



SpaceX Launch Records Dashboard

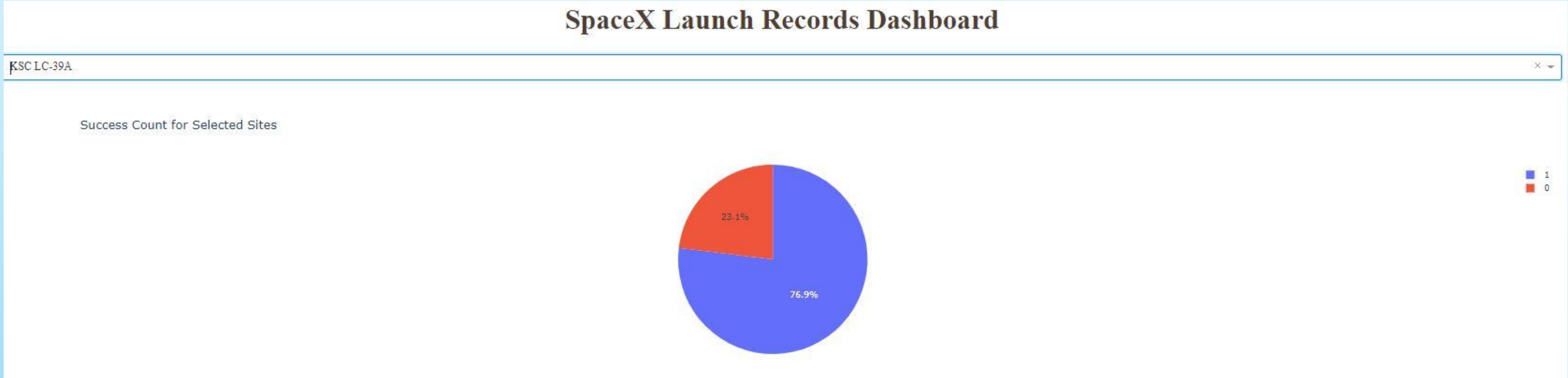
ALL

There are four launch sites.

‘*KSC LC-39A*’ has most percentage of successful missions of 41%.

‘*CCAFS SLC-40*’ has least percentage of successful missions of 12%.

PIECHART FOR THE LAUNCH SITE WITH HIGHEST LAUNCH SUCCESS RATIO

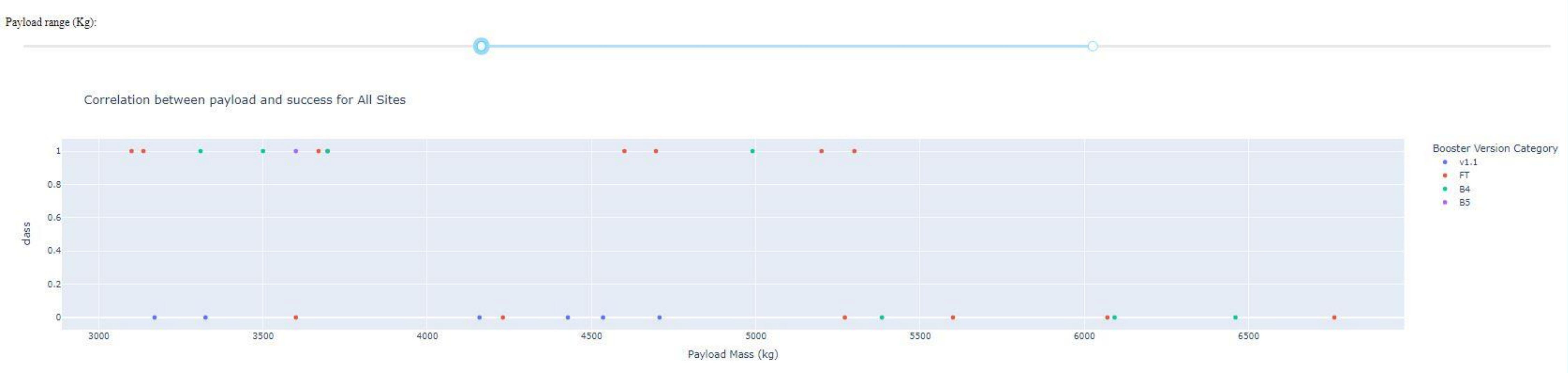


SpaceX Launch Records Dashboard

KSC LC-39A

'KSC LC-39A' has successful launches 76% of the time.

PAYLOAD VS. LAUNCH OUTCOME :ALL SITES



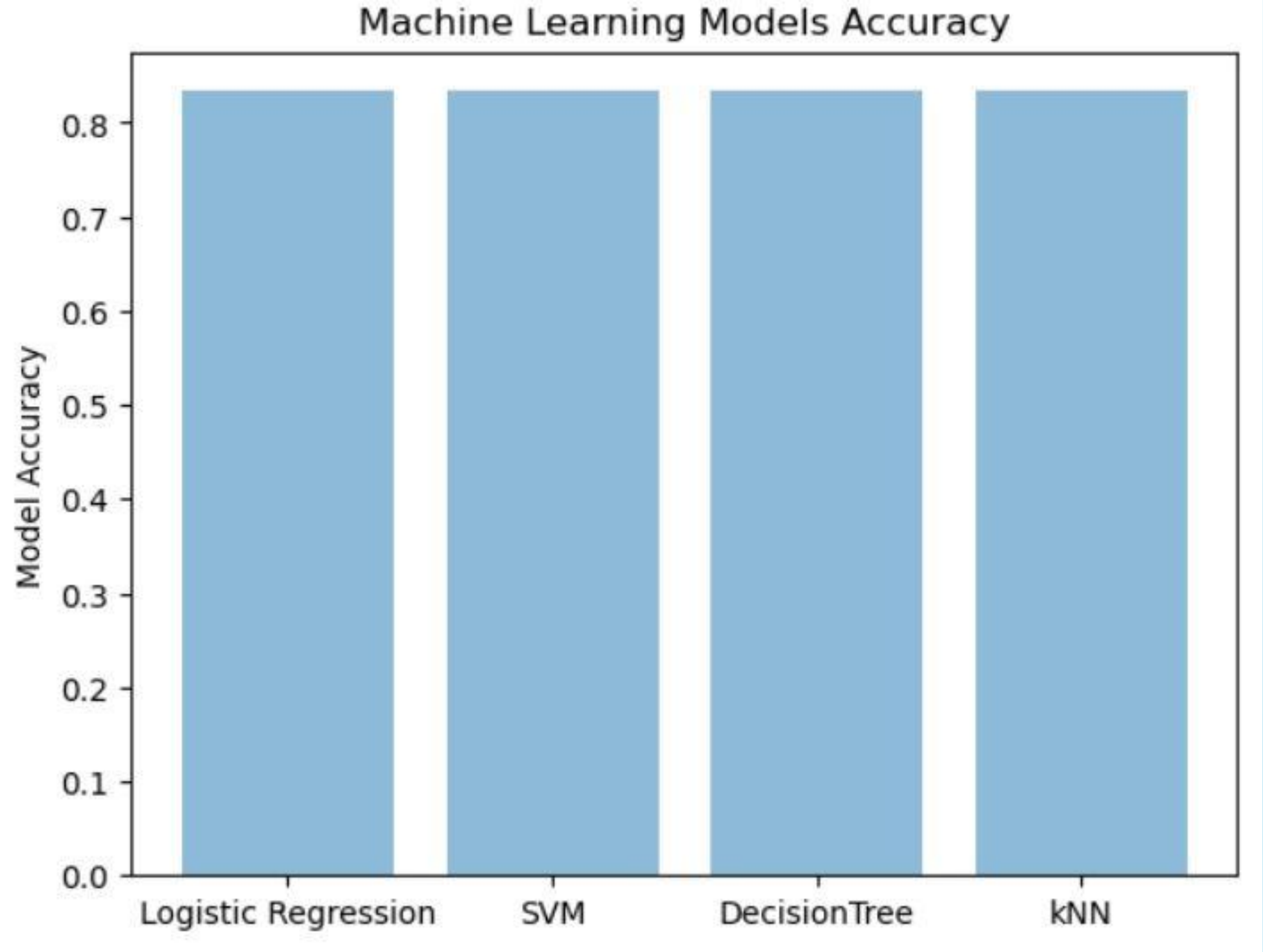
Payload range is selected to be between 2750 and 7000 kg.

All the sites are included

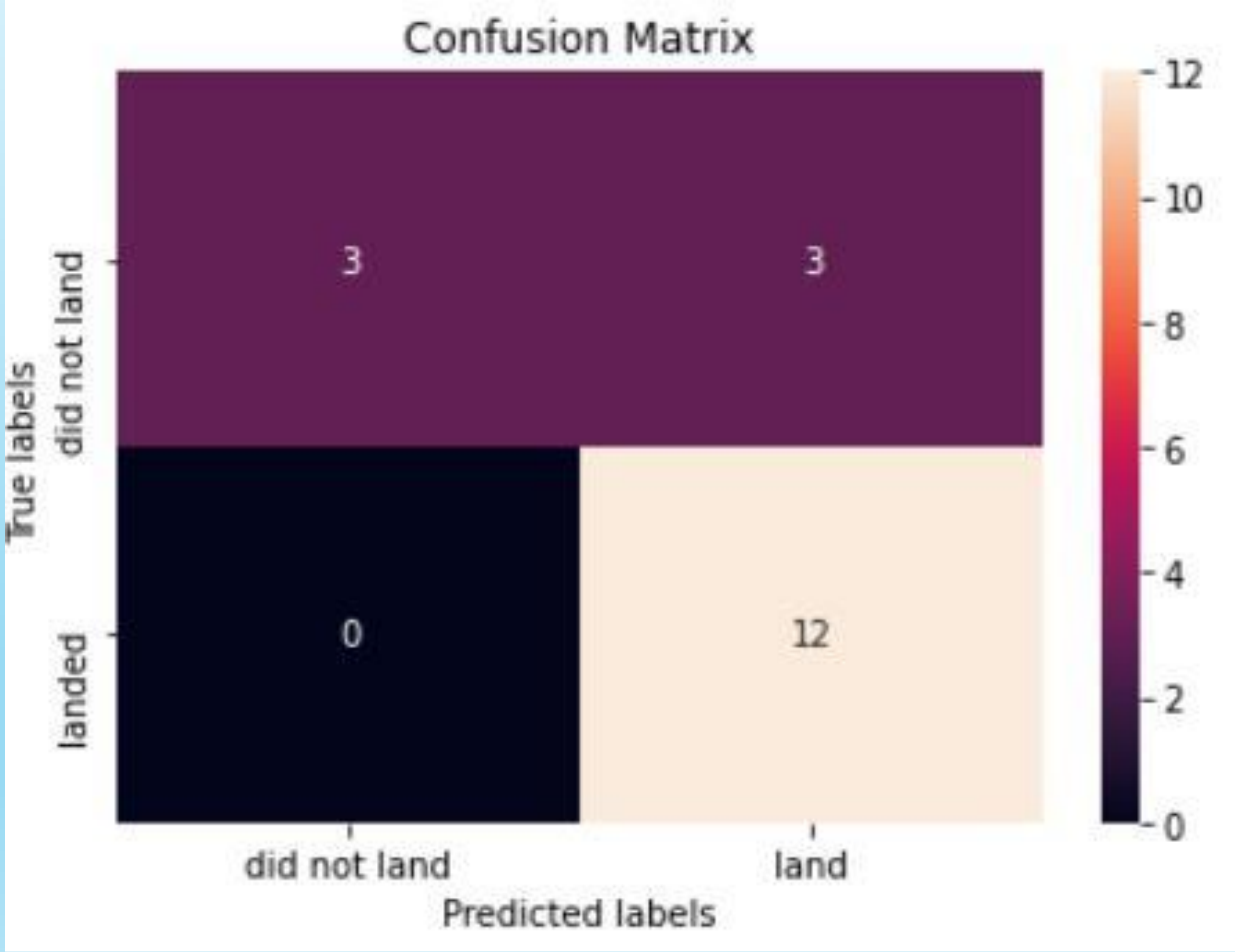
PREDICTIVE ANALYSIS (CLASSIFICATION)

CLASSIFICATION ACCURACY

- All models seem to have same accuracy of around 83% for this particular dataset



CONFUSION MATRIX - KNN



- Since all the models have same accuracy, confusion matrix of kNN has been displayed here.
- Precision is 80%. So, 80% of the time, landing success predicted by model comes true.
- Recall is 100%. So, model can predict all the successes.

CONCLUSION

Mission success rate has been steadily increasing since 2013.

In general, missions with payloads of more than 8000 kg have high success rate

SSO orbit missions have highest success rate where as GTO orbit missions have least success rates

All four ML models give same accuracy of about 83% to predict the mission success

APPENDIX

Github URL for the whole project :

https://github.com/ravirejo/DataScienceIBM_Course/tree/master