IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Sonia G>
<26/03/2022>

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix
- GitHub: -https://github.com/soniagoodluck/BM-Applied-Data-Science-Capstone

Executive Summary

Summary of methodologies

- Data collection using SPACEX REST API and Data wrangling
- Exploratory Data Analysis (EDA) performed using Python, and SQL queries
- Data visualization using Folium for mapping and Plotty dash for interactive dashboard
- Predictive analysis using Logistic regression, Support-vector machine, Decision tree and K-Nearest Neighbor

Summary of all results

- Exploratory Data Analysis (EDA) results
- Results of Data visualization with Folium and Plotty dash
- Predictive analysis results

Introduction

Project background

The cost of launching the Falcon 9 rocket is around 165 million dollars each on SpaceX's website, and saving is made by reusing the first stage. This project aims to explore the factors that can affect the successful landing of Falcon to determine the cost of a launch.

Objectives

- Find the correlation between the launch attributes and the outcome of the landing.
- Predict when a launch will land successfully.
- Find the best predictive model.

Methodology

Section 1

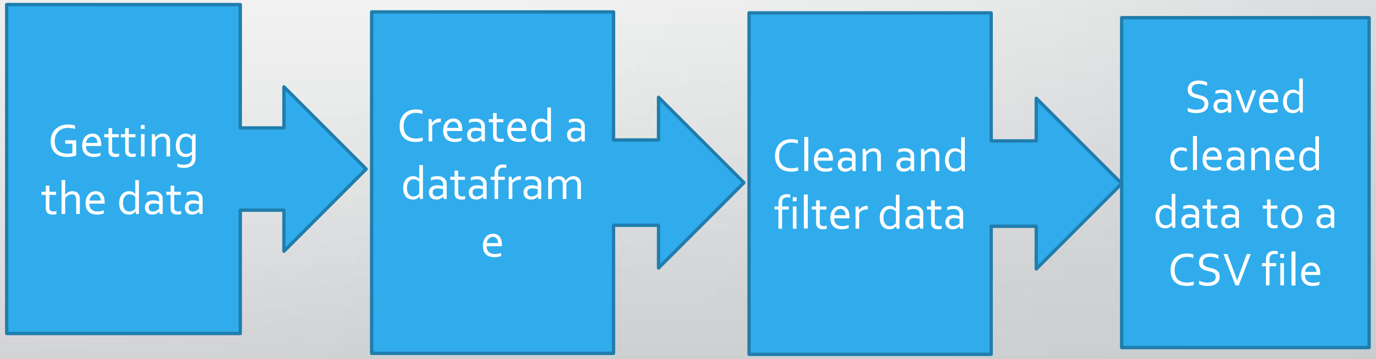
Methodology

Executive Summary

- Data collection methodology:
  - Request to the SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - Dealt with missing values by replacing with the mean and also converted categorical column into numerical value.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Standardize the data, split the data into train and test set then train the models and then choose the best model.

Data Collection

The dataset were collected from two different sources using request library. The first dataset was collected from the SpaceX API and the second with the help of BeautifulSoup the dataset was extract from Wikipedia. The Data were cleaned and saved into individual CSV file



Getting the data

Created a dataframe

Clean and filter data

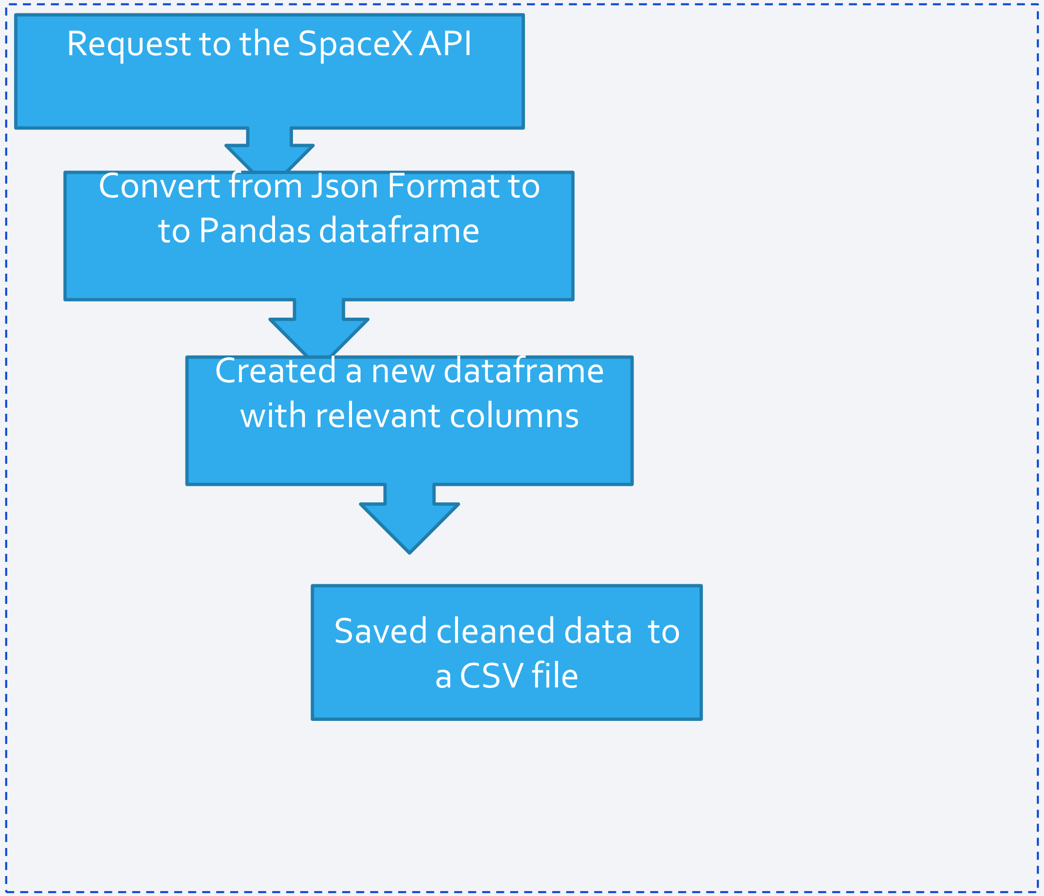
Saved cleaned data to a CSV file

Data Collection – SpaceX API

- Request to the SpaceX API to get data containing information about past lunches from - https://api.spacexdata.com/v4/launches/past
- Converted data from Json format to Pandas dataframe using .json_normalize().
- Created a new dataframe with only columns needed such as 'rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc'
- Performed data cleaning on the new dataframe.
- Saved cleaned data to a CSV file.

GitHub URL-

https://github.com/soniagoodluck/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Request to the SpaceX API

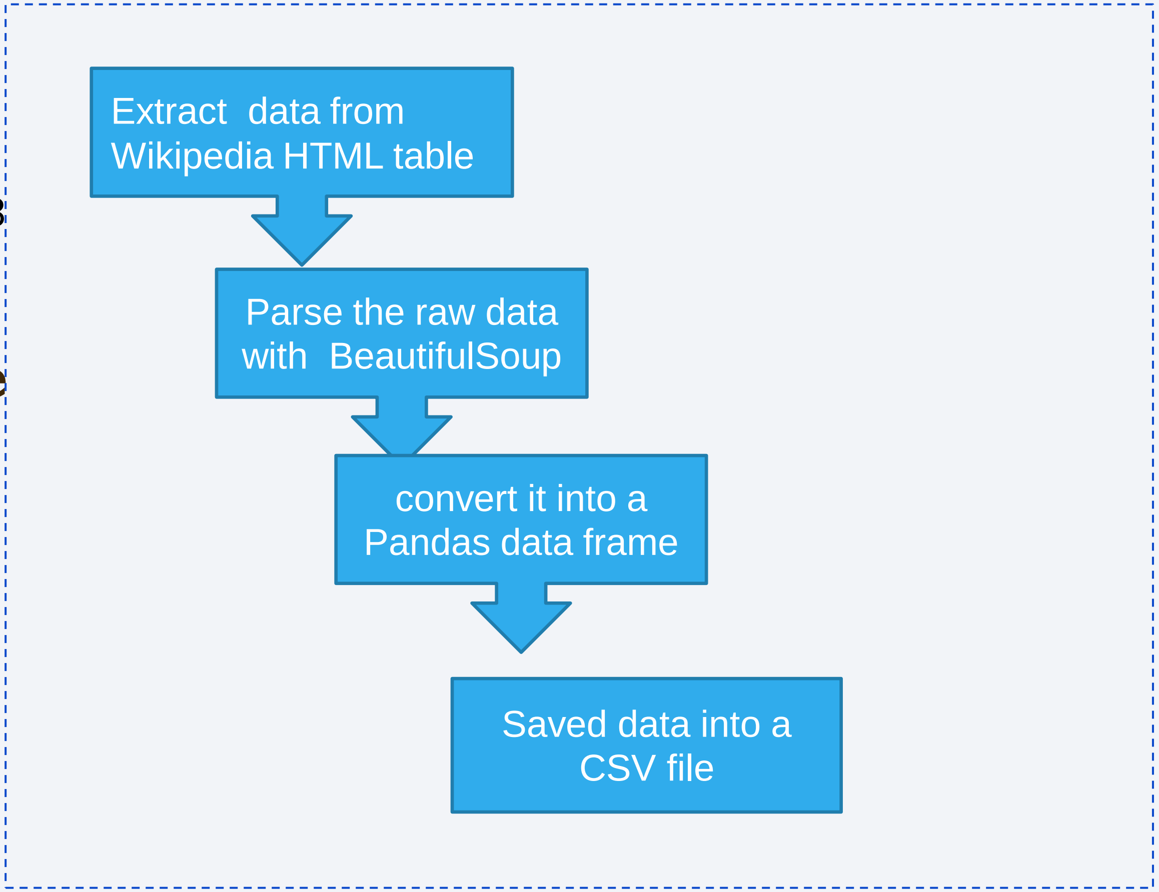
Convert from Json Format to to Pandas dataframe

Created a new dataframe with relevant columns

Saved cleaned data to a CSV file

Data Collection - Scraping

- Got historical launch data from the Wikipedia using web scraping - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- Beautiful Soup was used to parse the raw data by helping to organize and format it to a JSON structure.
- Created a Pandas DataFrame.
- Saved data into a CSV file.
- GitHub URL- https://github.com/soniagoodluck/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb


Extract data from Wikipedia HTML table

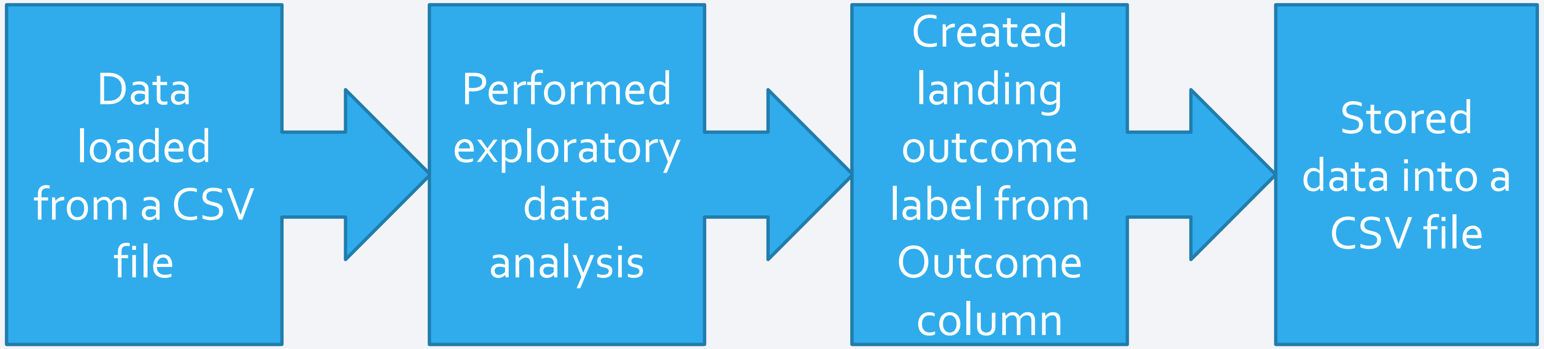
Parse the raw data with BeautifulSoup

convert it into a Pandas data frame

Saved data into a CSV file

Data Wrangling

- The data was load into Jupiter notebook from a CSV file, we carried out exploratory data analysis (EDA) to determine the dataset structure by creating a landing outcome label from Outcome column.



Data loaded from a CSV file
Performed exploratory data analysis
Created landing outcome label from Outcome column
Stored data into a CSV file

- GitHub URL - https://github.com/soniagoodluck/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Performed exploratory Data Analysis and Feature Engineering using Pandas, Matplotlib and Seaborn libraries
- Scatterplot chart to see how the relationship between the variables will affect the outcome of landing.
- Bar chart was used to find the relationship between the success rate of each variables.
- Line chart was used to view the yearly trend of success rate.

EDA with SQL

By connecting to IBM Db2 database using Python we were able to run some SQL queries showing-

- The names of the unique launch sites in the space mission
- 5 records where launch sites begin with the string 'CCA’.
- Total payload mass carried by boosters launched by NASA (CRS)
- The average payload mass carried by booster version F9 v1.1
- Date when the first successful landing outcome in ground pad was acheived.
- Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Total number of successful and failure mission outcomes
- Names of the booster_versions which have carried the maximum payload mass. Use a subquery
- The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- 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

GitHub URL- https://github.com/soniagoodluck/IBM-Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- Added a circle with popup name indicating each launch site.
- Added a columns with color label related to the outcome of landing (green for a successfully landing and red for a failed one).
- Create a marker cluster object.
- Added the color-coded map icons to the marker cluster using the coordinates of the launch site.
- Calculated and added lines that shows the distances between a launch site to its nearest proximities (Coast, City, Highway, Railway).
- By adding these objects, we can virtualize the distances between a launch site to its nearest proximities (Coast, City, Highway, Railway).
- GitHub URL - https://github.com/soniagoodluck/IBM-Applied-Data-Science-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

The dashboard app contains a pie chart and scatterplot.

- Graphs

  - Pie chart showing the total of launches by each site and all site.

  - Displays relative proportions of multiple classes of data.

  .- Size of circle can be made proportional to the total quantity it represents.

- Scatterplot

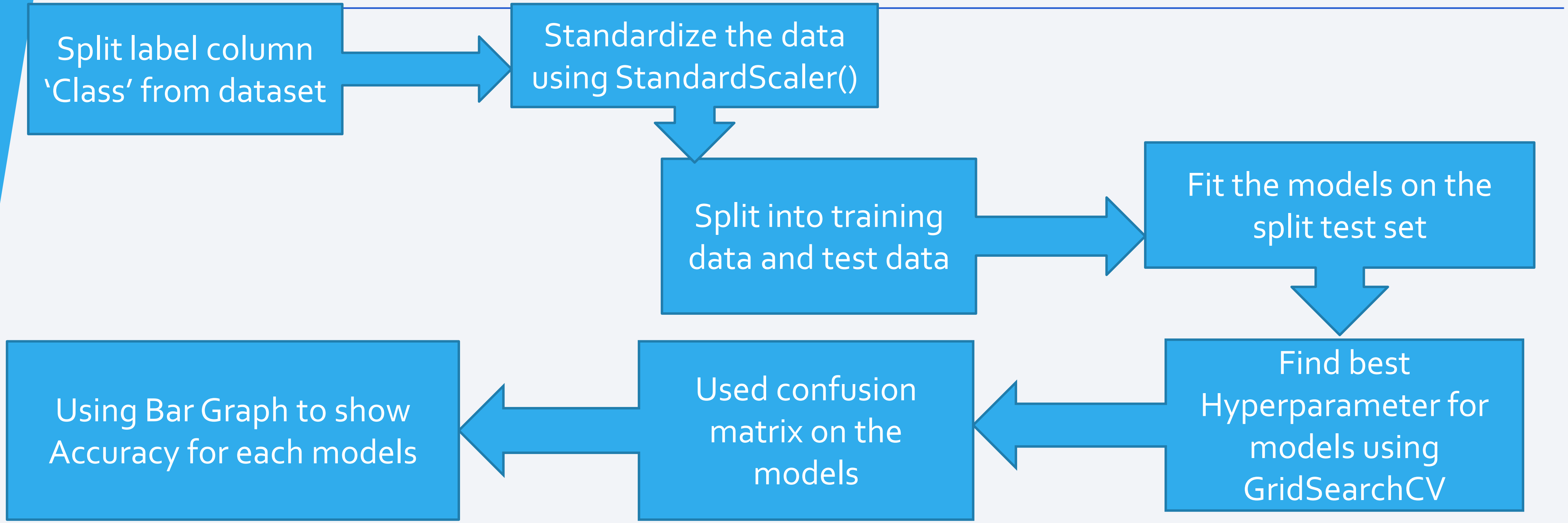
  - Shows the relationship between the Outcomes and Payload mass (kg) by different boosters.

  - helps to determine the range of data flow i.e. maximum or minimum value.

  - The best method to show a noon-linear pattern and observation and reading are straightforward.

- GitHub URL- https://github.com/soniagoodluck/IBM-Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)



Split label column ‘Class’ from dataset

Standardize the data using StandardScaler()

Split into training data and test data

Fit the models on the split test set

Find best Hyperparameter for models using GridSearchCV

Used confusion matrix on the models

Using Bar Graph to show Accuracy for each models

- GitHub URL- https://github.com/soniagoodluck/IBM-Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

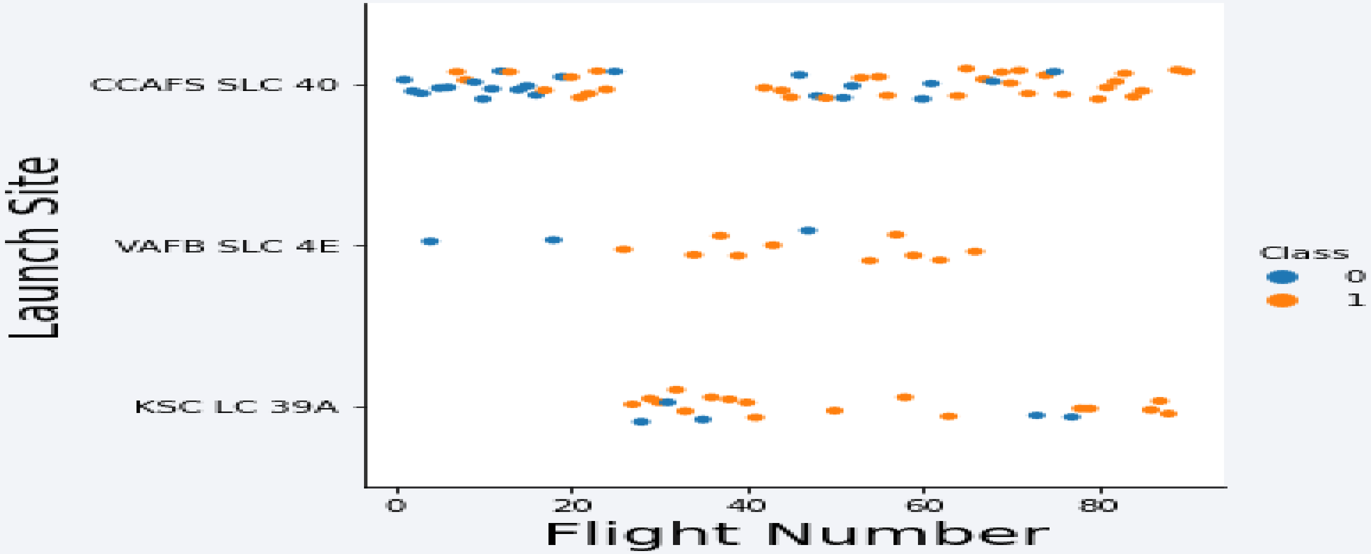
Results

- Exploratory data analysis results shows that payload mass has an impact in the landing outcome, along with the orbit and booster versions.
- Folium map displays the proximities of launch sites to nearby coastlines, highway and railways.
- Predictive analysis results predicted the Algorithm to performs best is Decision Tree with a score of 0.875.

Section 2

Insights drawn from EDA

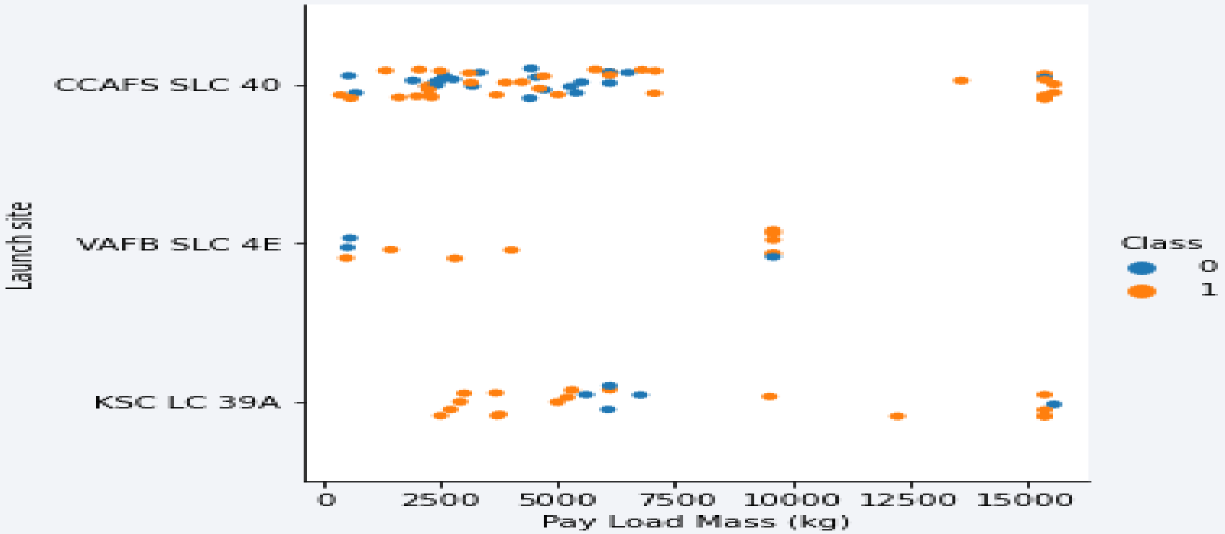
Flight Number vs. Launch Site



Class 0 – (Blue color) represents unsuccessful launch while Class 1 (orange color) represents successful

- There is an increase in success rate over time (indicated in Flight Number). KSC LC-39A seems to be the most successful site. CCAFS SLC 40 appears to have the highest numbers of flights.

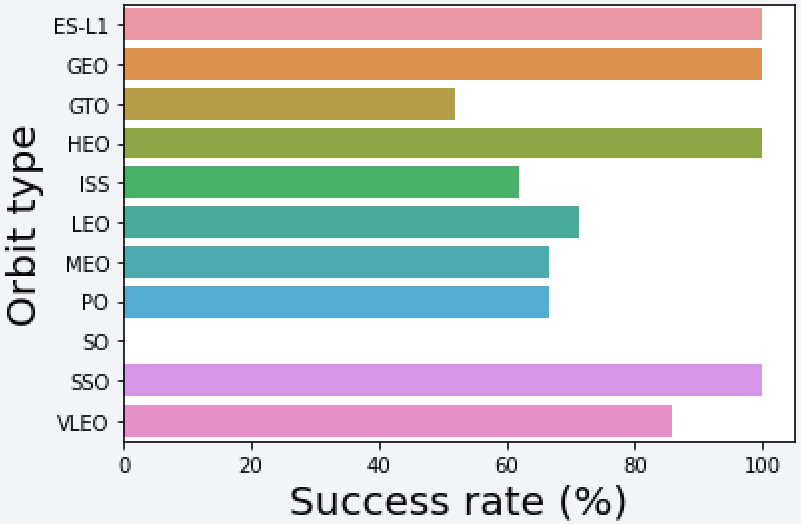
Payload vs. Launch Site



Class 0 – (Blue color) represents unsuccessful launch while Class 1 (orange color) represents successful

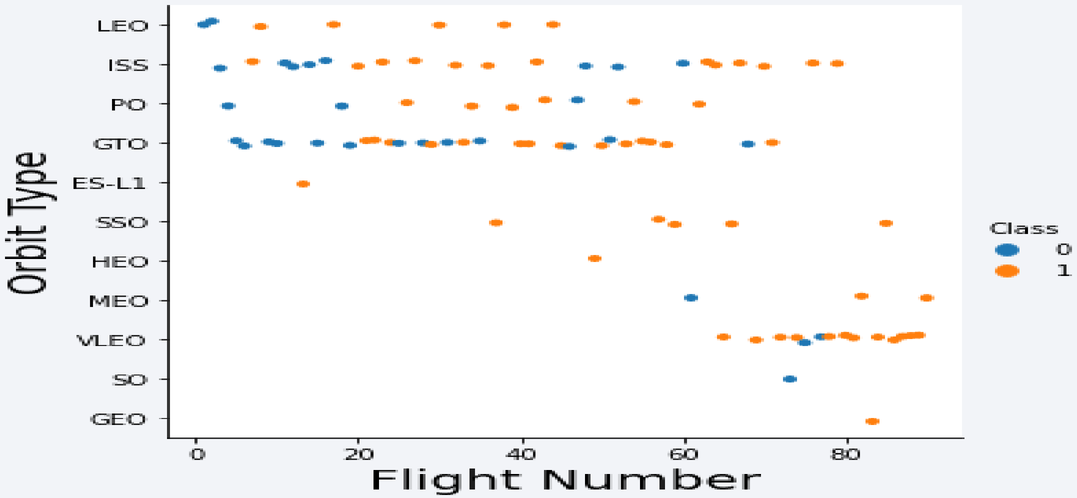
- it seems the launches with higher payloads were more successful.

Success Rate vs. Orbit Type



- The GEO, ES-L1, SOO and HEO orbits have the best success while the GTO has the lowest rate.

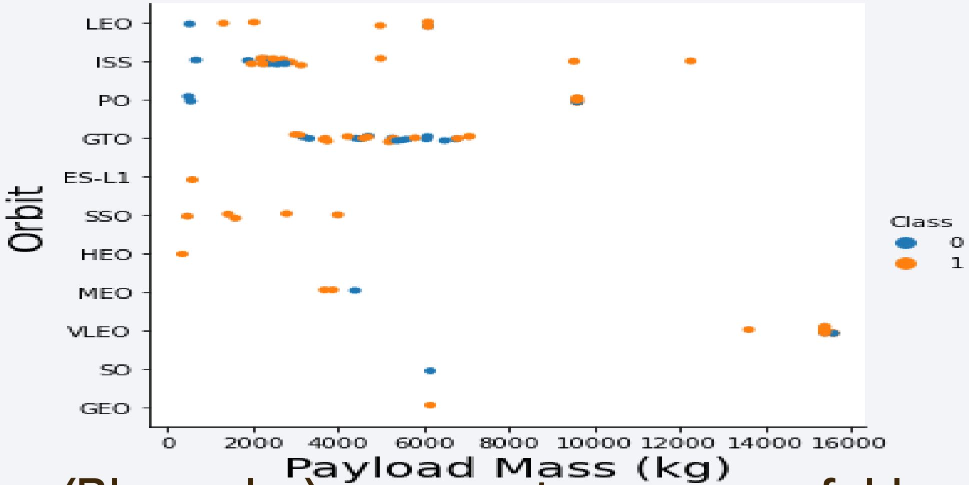
Flight Number vs. Orbit Type



Class 0 – (Blue color) represents unsuccessful launch while Class 1 (orange color) represents successful

- Show the screenshot of the scatter plot with explanations

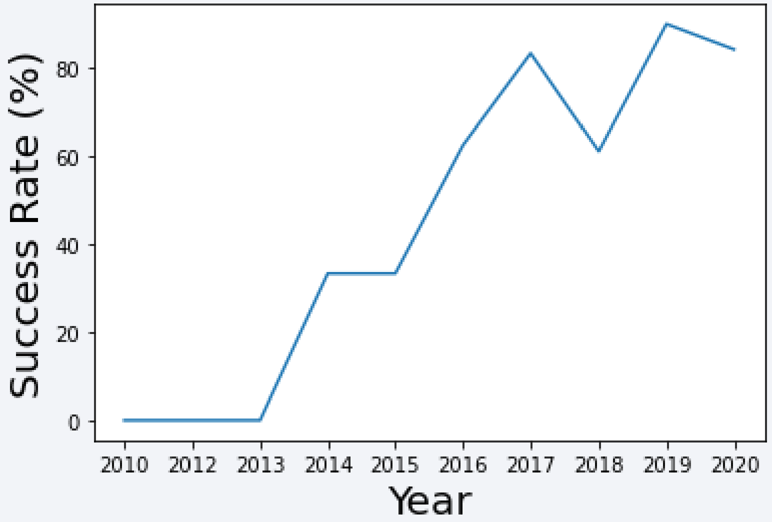
Payload vs. Orbit Type



Class 0 – (Blue color) represents unsuccessful launch while Class 1 (orange color) represents successful

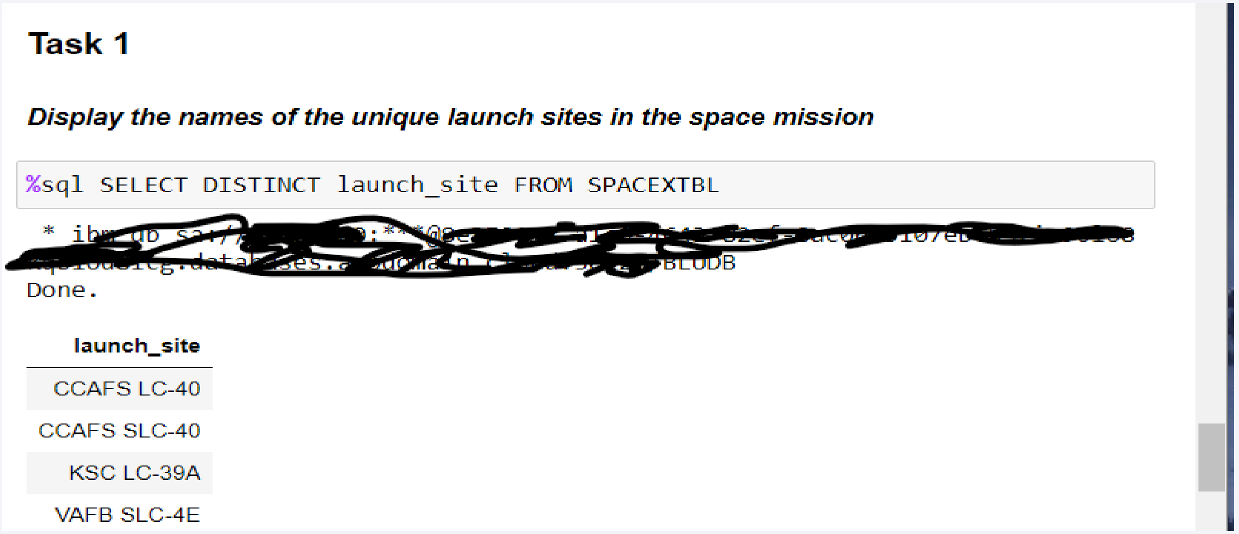
- The success rate is more with heavy payloads in PO, LEO and ISS and with lower payloads in SSO, HEO, ES-L1.
- GTO has no relationship with payload mass.

Launch Success Yearly Trend



- The success rate increased up till 2017, then a decline till 2019 and after there was a spike till the end of 2021

All Launch Site Names



Task 1

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

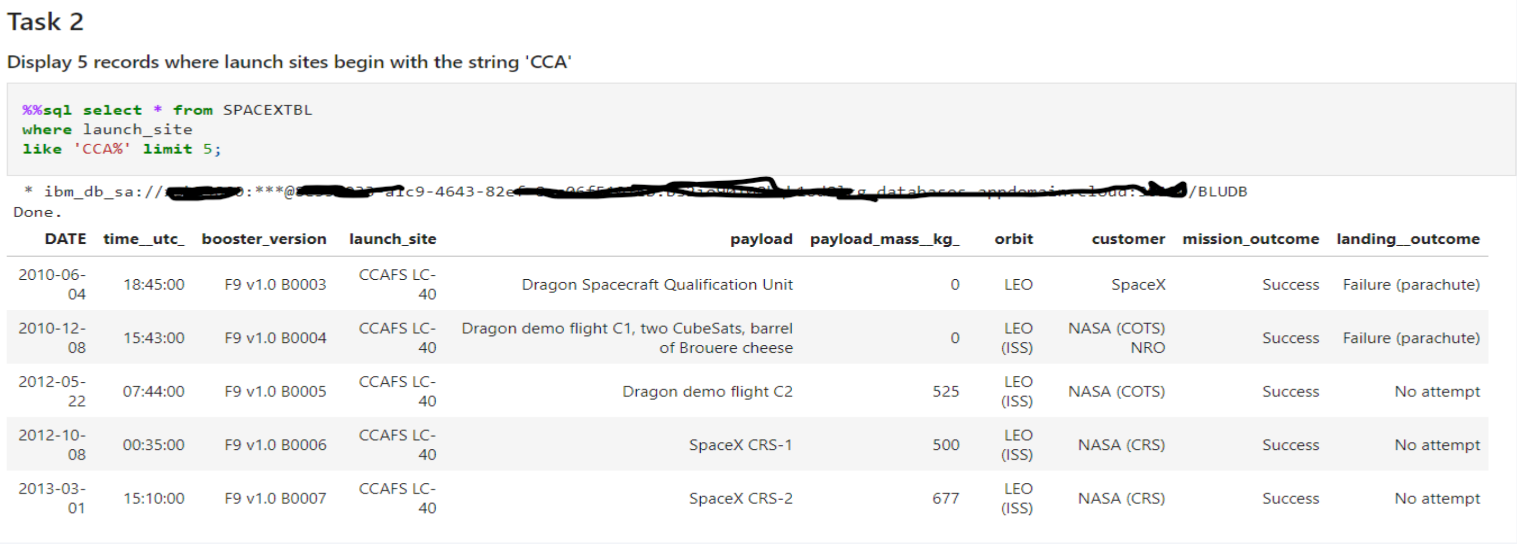
```
%sql SELECT DISTINCT launch_site FROM SPACEXTBL
```

```
[redacted]
Done.
```

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

- Using the word DISTINCT in the query shows the names of the unique launch sites in the space mission and there are 4 unique launch site.

Launch Site Names Begin with 'CCA'



Task 2

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

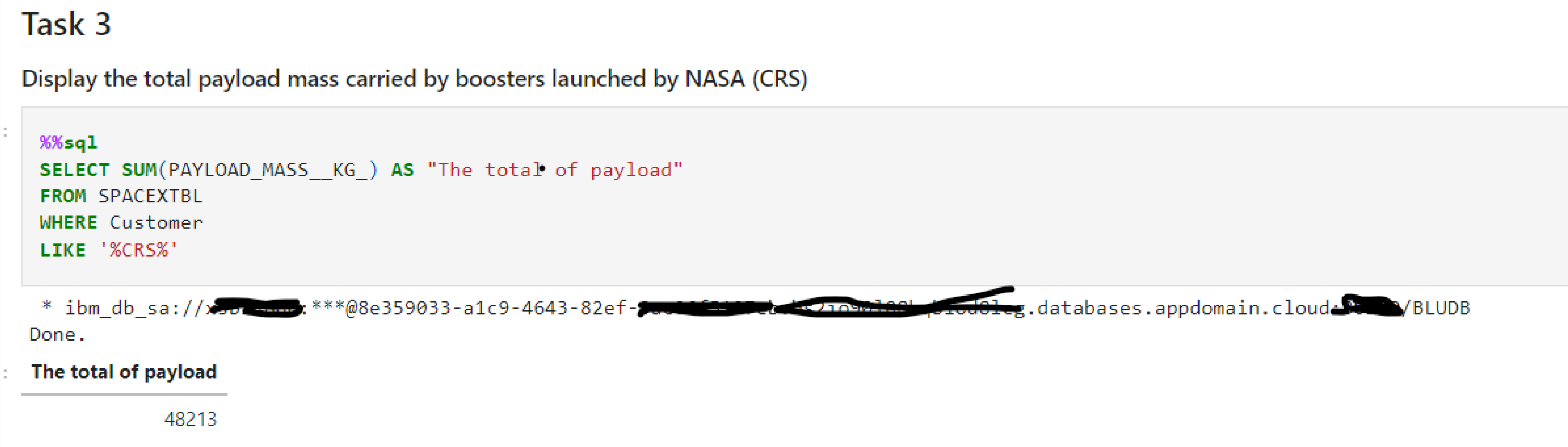
```
%%sql select * from SPACEXTBL
where launch_site
like 'CCA%' limit 5;
```

```
 * ibm_db_sa://[redacted]:***@[redacted]a1c9-4643-82e[redacted]/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

- Showing the first five entries in database where launch sites begin with the string 'CCA’ with the aid of LIMIT 5, LIKE operator and the percent sign %.

Total Payload Mass



Task 3

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

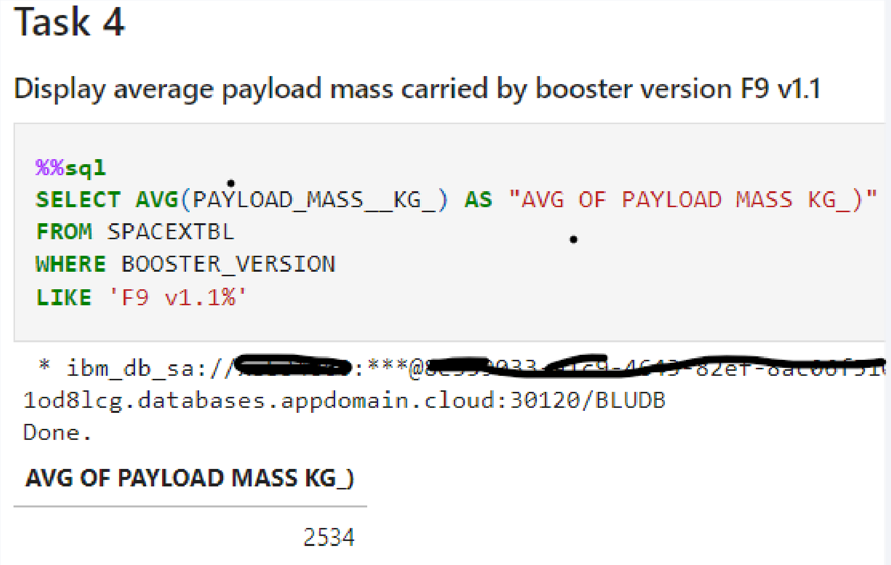
```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS "The total of payload"
FROM SPACEXTBL
WHERE Customer
LIKE '%CRS%'
```

```
 * ibm_db_sa://[redacted]:***@8e359033-a1c9-4643-82ef-[redacted].databases.appdomain.cloud[redacted]/BLUDB
Done.
```

The total of payload
48213

- Displaying the total payload mass carried by boosters launched by NASA (CRS) with the SUM() function to calculate and the WHERE clause helps to filter the condition.

Average Payload Mass by F9 v1.1



Task 4

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

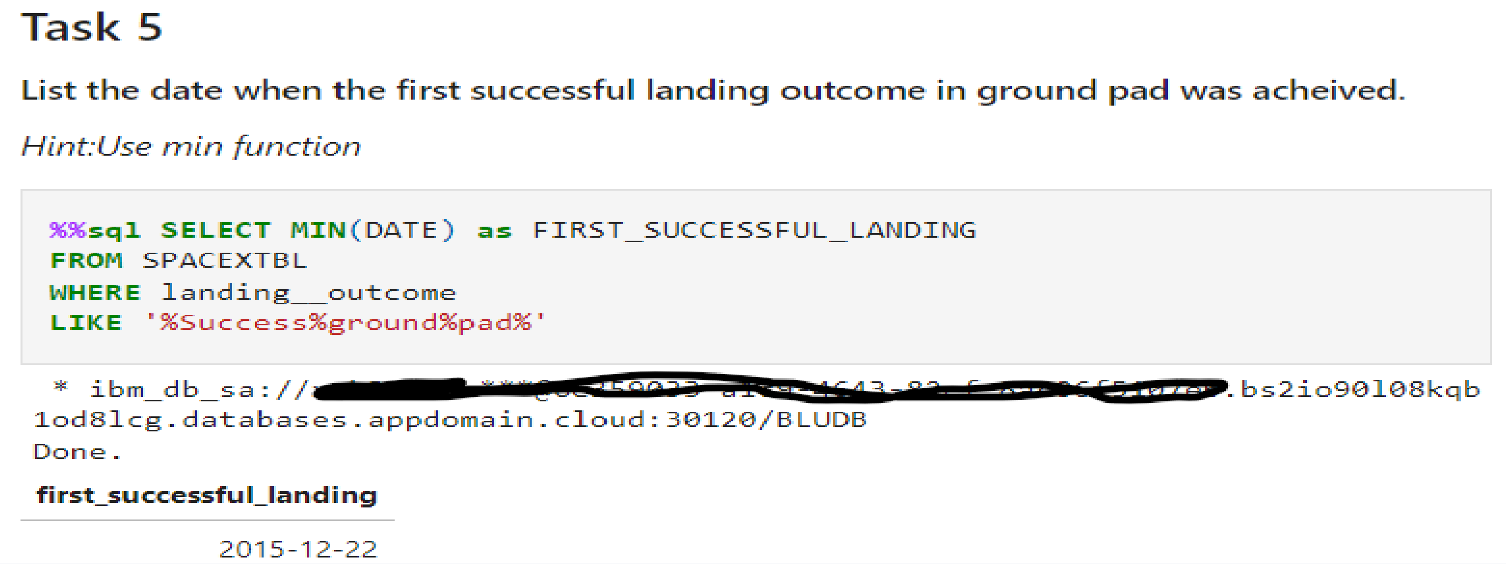
```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS "AVG OF PAYLOAD MASS KG_)"
FROM SPACEXTBL
WHERE BOOSTER_VERSION
LIKE 'F9 v1.1%'
```

```
 * ibm_db_sa://[redacted]:***@[redacted]
1od8lcg.databases.appdomain.cloud:30120/BLUDB
Done.
```

AVG OF PAYLOAD MASS KG_)
2534

- Displaying average payload mass carried by booster version F9 v1.1 using the AVG function to calculate the average.

First Successful Ground Landing Date



Task 5

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

Hint:Use min function

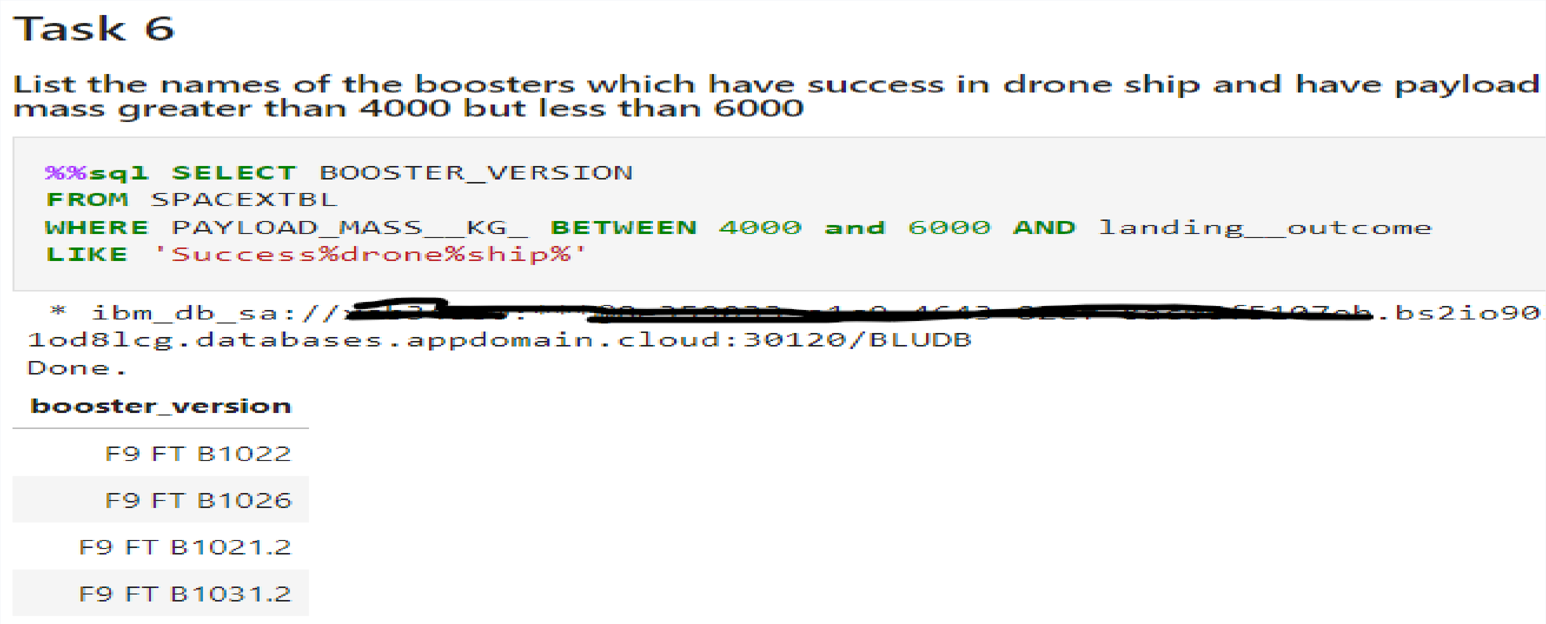
```
%%sql SELECT MIN(DATE) as FIRST_SUCCESSFUL_LANDING
FROM SPACEXTBL
WHERE landing__outcome
LIKE '%Success%ground%pad%'
```

```
 * ibm_db_sa://[redacted].bs2io90l08kqb
1od8lcg.databases.appdomain.cloud:30120/BLUDB
Done.
```

first_successful_landing
2015-12-22

- Listing the date when the first successful landing outcome in ground pad was achieved with the MIN() function.

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

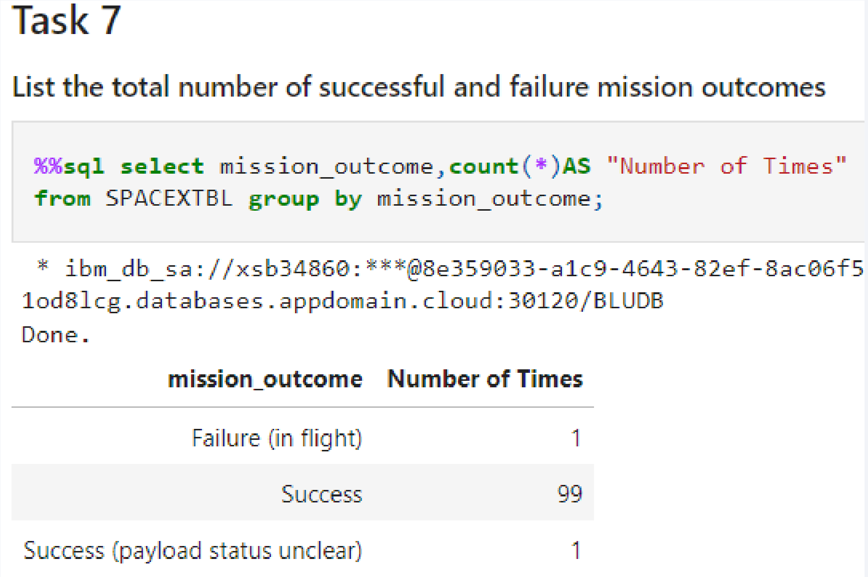
```
%%sql SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000 AND landing__outcome
LIKE 'Success%drone%ship%'
```

```
 * ibm_db_sa://[redacted].bs2io90
1od8lcg.databases.appdomain.cloud:30120/BLUDB
Done.
```

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

- Listing the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 with the c of WHERE clause, the AND Between and LIKE clause.

Total Number of Successful and Failure Mission Outcomes



Task 7

List the total number of successful and failure mission outcomes

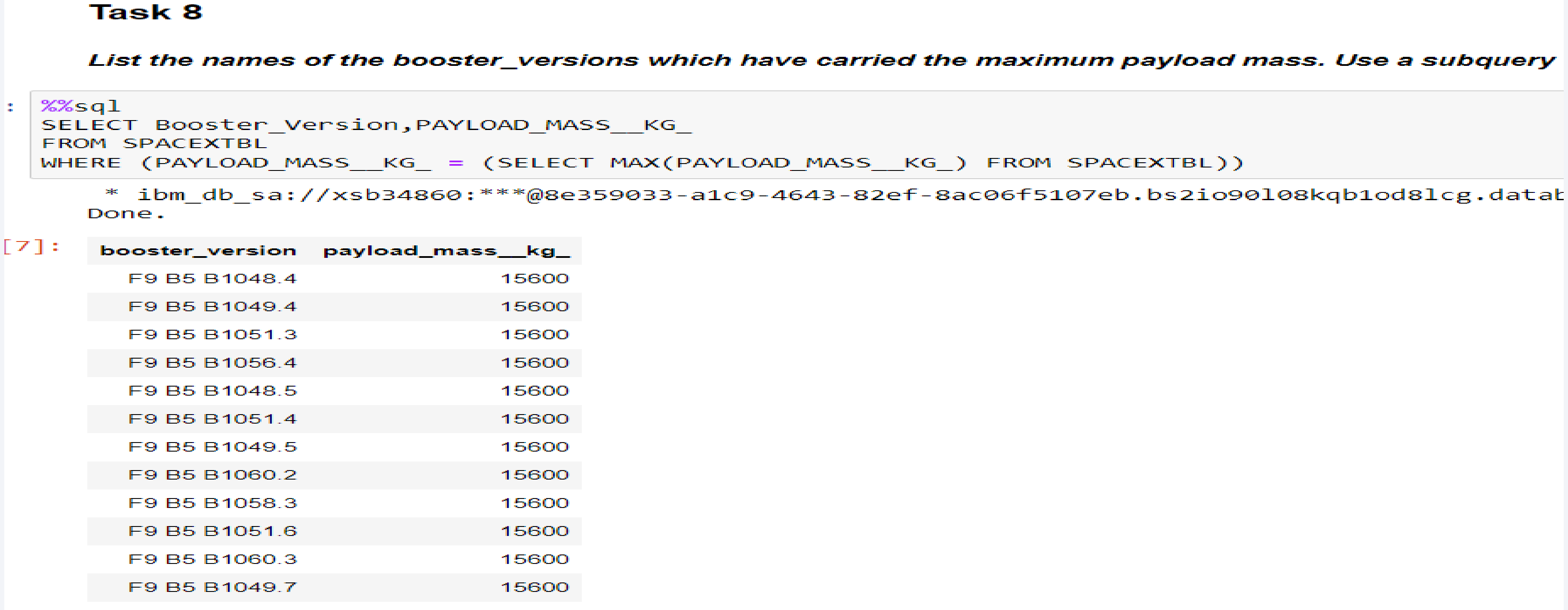
```
%%sql select mission_outcome,count(*)AS "Number of Times"
from SPACEXTBL group by mission_outcome;
```

```
 * ibm_db_sa://xsb34860:***@8e359033-a1c9-4643-82ef-8ac06f5
1od8lcg.databases.appdomain.cloud:30120/BLUDB
Done.
```

mission_outcome	Number of Times
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- This query returns the total number of successful and failure mission outcomes.

Boosters Carried Maximum Payload



Task 8

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

```
%%sql
SELECT Booster_Version,PAYLOAD_MASS__KG_
FROM SPACEXTBL
WHERE (PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL))
```

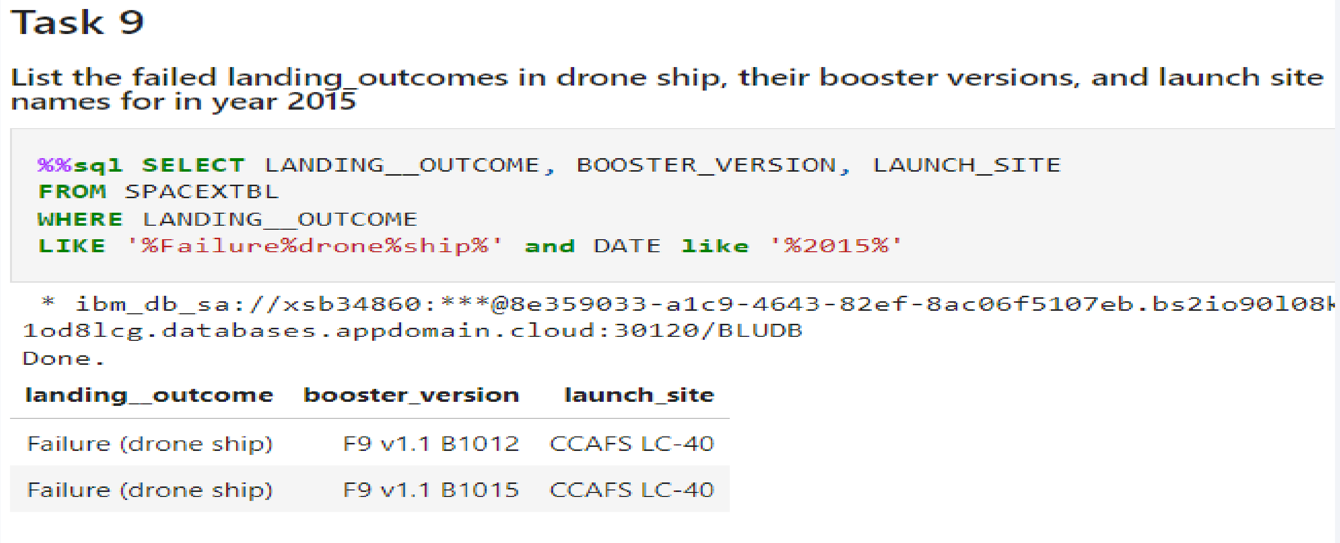
```
 * ibm_db_sa://xsb34860:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.datab
Done.
```

[7]:

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

- Listing the names of the booster which have carried the maximum payload mass of 15600 kg.

2015 Launch Records



Task 9

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

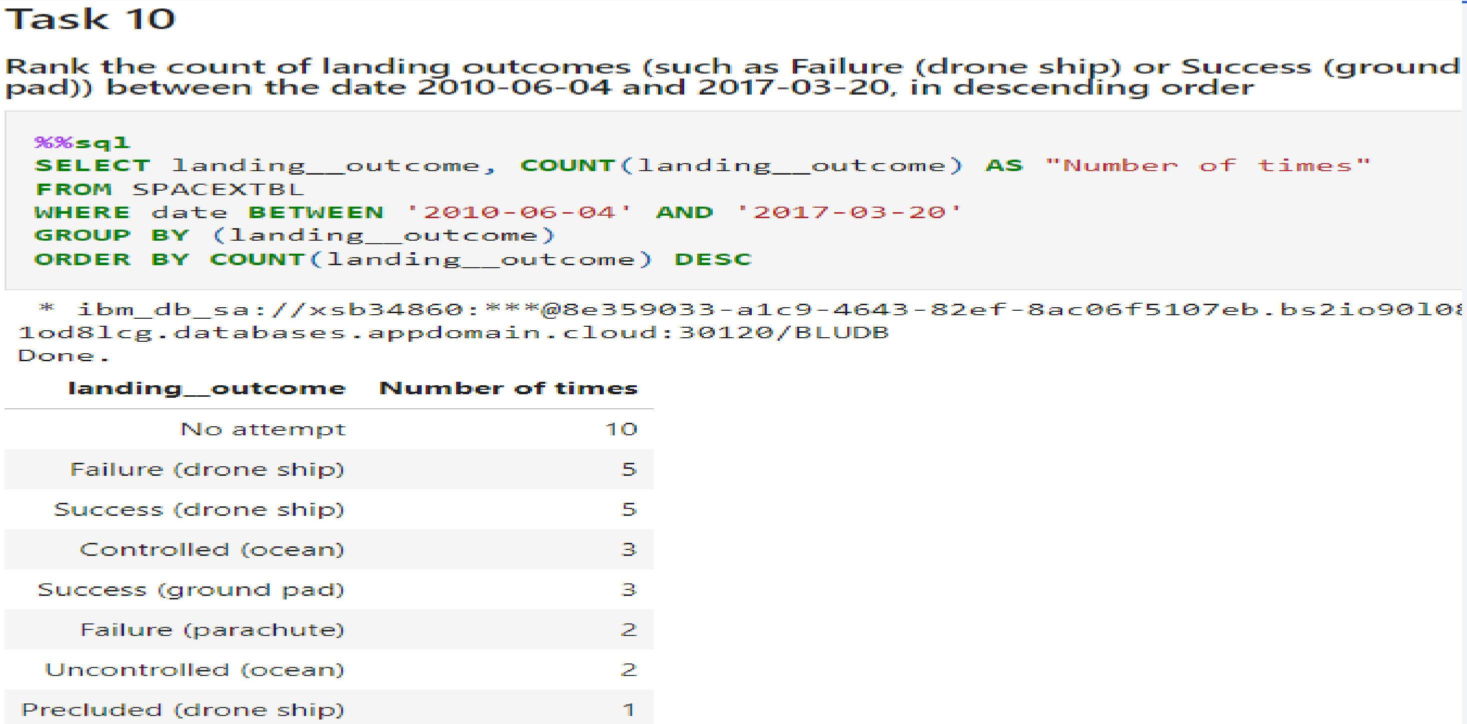
```
%%sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING__OUTCOME
LIKE '%Failure%drone%ship%' and DATE like '%2015%'
```

```
 * ibm_db_sa://xsb34860:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08k
1od8lcg.databases.appdomain.cloud:30120/BLUDB
Done.
```

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

- Listing the failed landing outcomes for the year 2015 on 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

```
%%sql
SELECT landing__outcome, COUNT(landing__outcome) AS "Number of times"
FROM SPACEXTBL
WHERE date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY (landing__outcome)
ORDER BY COUNT(landing__outcome) DESC
```

```
 * ibm_db_sa://xsb34860:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90108
1od8lcg.databases.appdomain.cloud:30120/BLUDB
Done.
```

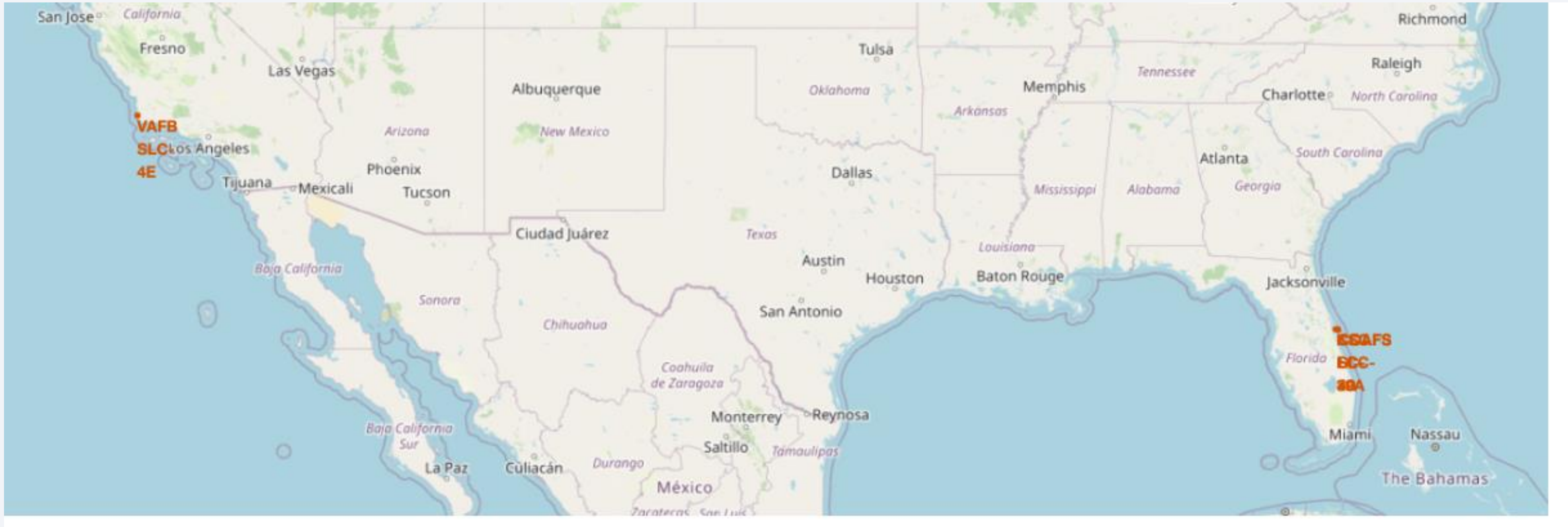
landing__outcome	Number of times
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- This query returns a list 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

Section 3

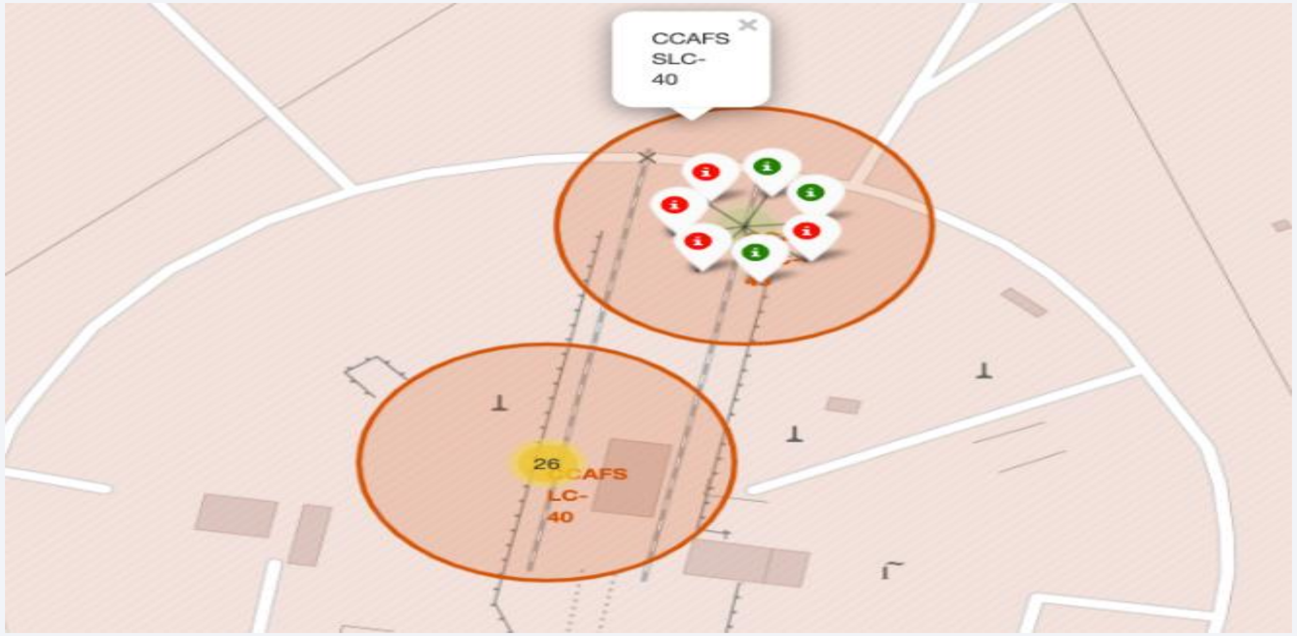
Launch Sites Proximities Analysis

Map of Launch sites



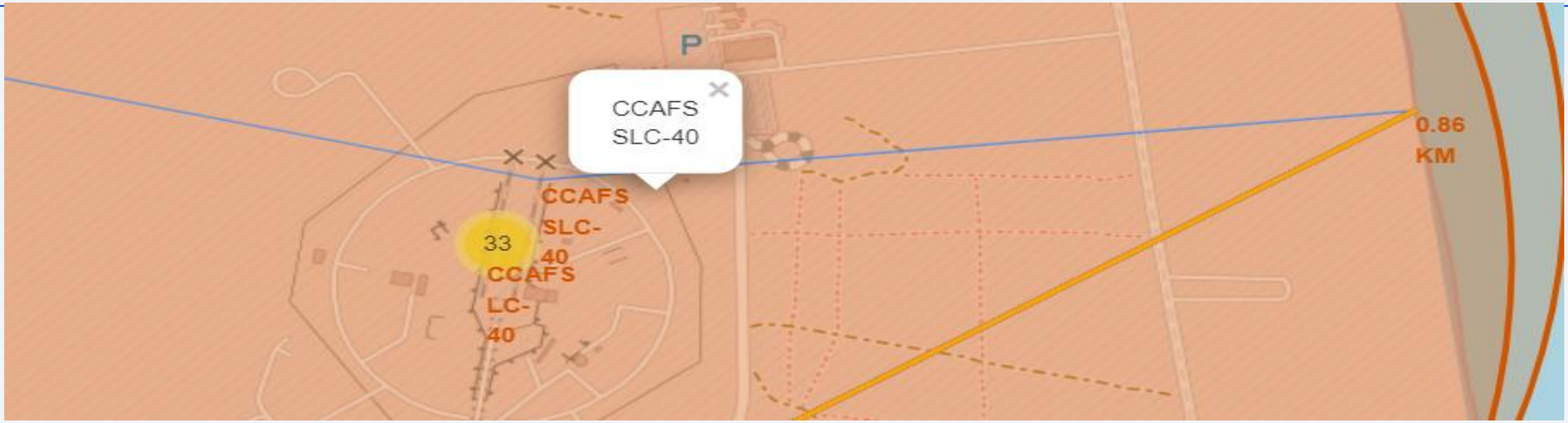
- The SpaceX launch site is in the United State of America near the ocean.

Color-Coded Launch Markers



- Clusters on Folium map can be clicked on to display each successful (green icon) and failed landing (red icon).

The distances between CCCAFS SLC-40 launch site to the coastline

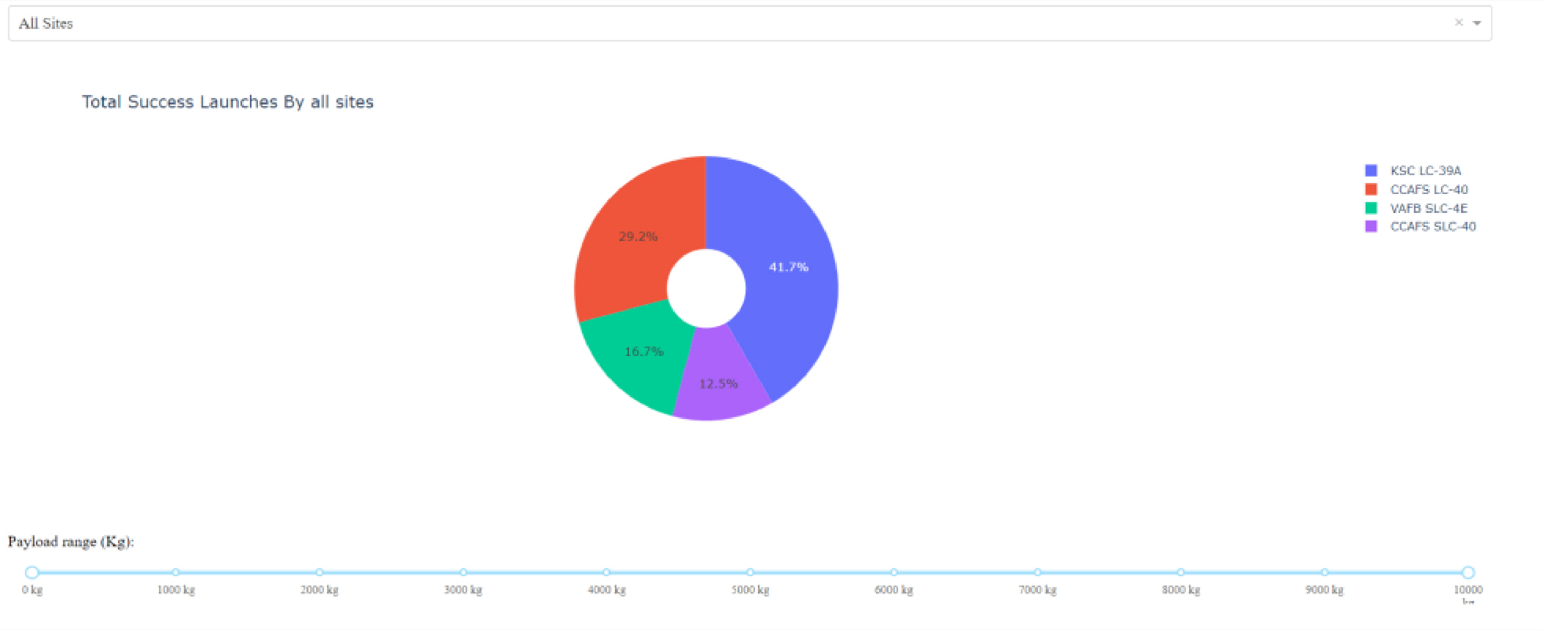


- CCCAFS SLC-40 is 0.86km from the coastline.

Section 4

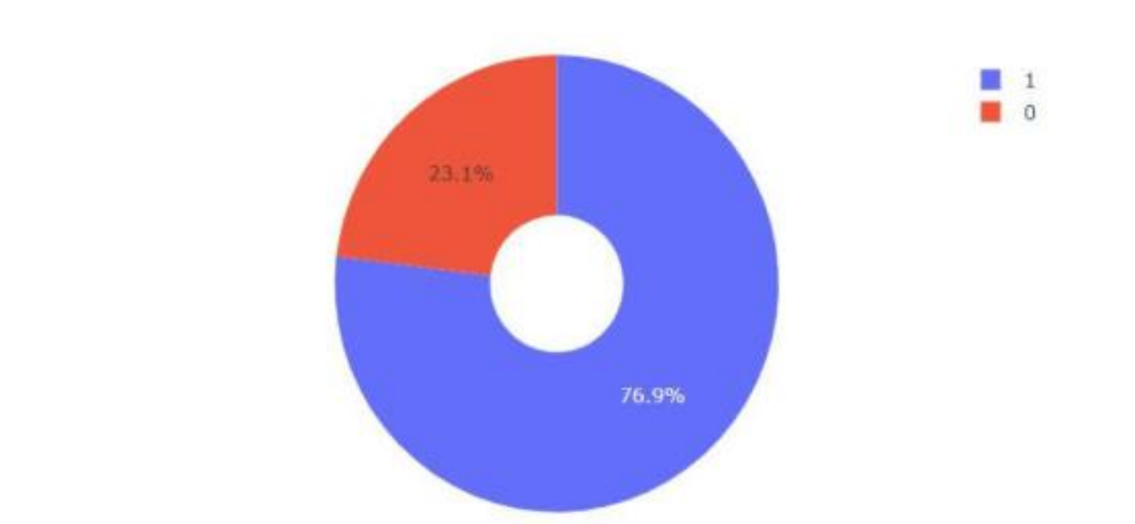
Build a Dashboard with Plotly Dash

Launch Success across all site



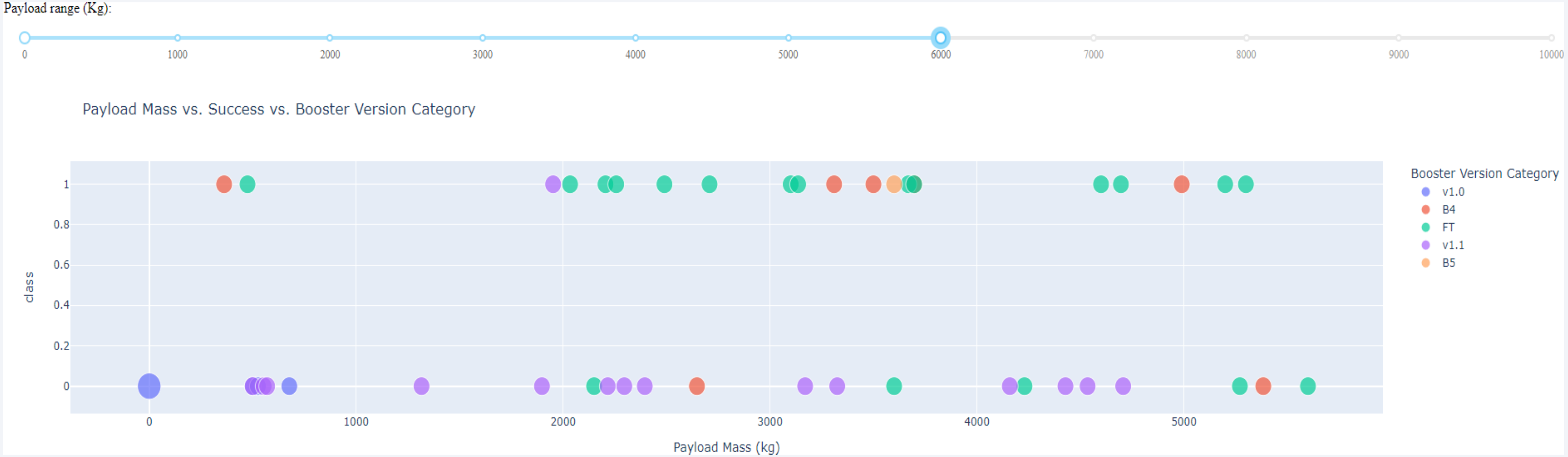
The KSC LC-39A site has the highest success counts, with 41.7% followed by CCAFS LC-40 with 29.2% and CCATS SLC-40 has the least with 12.5%

The Success Rate for KSC LC-39A site



- The KSC LC-39A site scored a success rate of 76.9 and a failure rate of 23.1%

Payload vs. Launch Outcome scatter plot for all sites



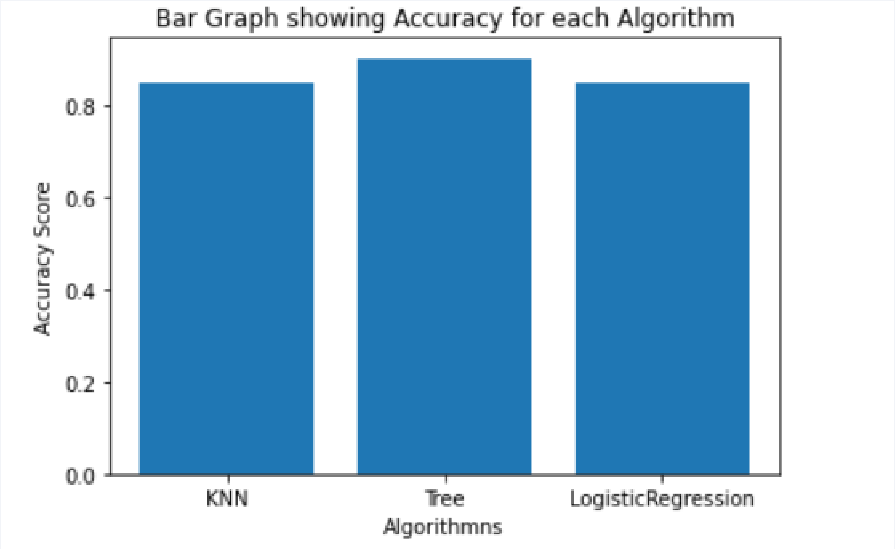
Payload range (Kg):

- The plot shows booster version FT with the highest success rate in comparison to other booster versions.

Section 5

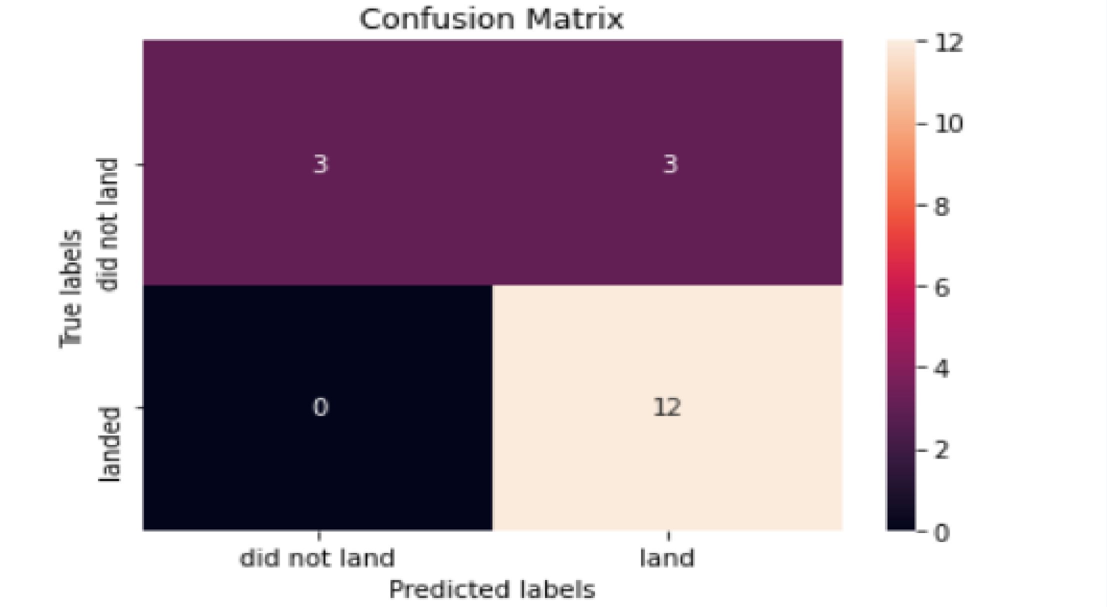
Predictive Analysis (Classification)

Classification Accuracy



- The model that has the highest classification accuracy is Tree with 0.875.

Confusion Matrix



- 12 landed outcomes were correctly predicted.
- 3 failed outcomes are correctly predicted.
- 3 more failed outcomes are falsely predicted.

Conclusions

- KSC LC-39A had the highest success rate in launching compared to other launch sites.
- Higher weighted payload performed better than the lower payloads.
- Orbit GEO, HEO, SSO, ES-L1 has best success rates.
- Payload mass, booster version and orbit combined has a great effect on the outcome of landing
- Launch success rates increase with time.
- The decision tree Algorithm was the best Machine learning for the dataset.

Appendix

- GitHub Link: https://github.com/soniagoodluck/IBM-Applied-Data-Science-Capstone

Thank you!