Data Science Capstone Project

Minhajur Rahman Khan

30-Aug-2021

Outline

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

Executive Summary

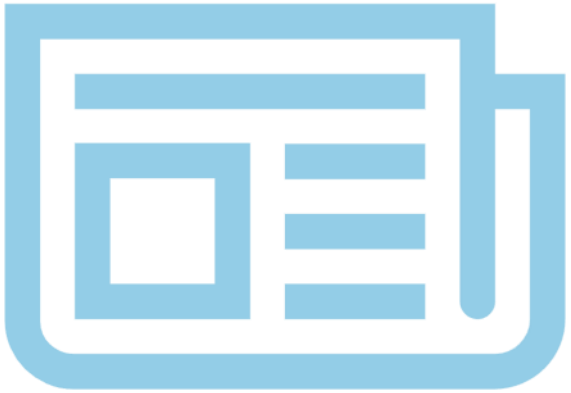
- Used different data collection methods to collected data (direct data files from web, web scraping, JSON web API).
- Performed exploratory data analysis (EDA) using visualization and SQL.
- Performed predictive analysis using different classification models (logistic regression, decision tree, svm, knn).
- Got model accuracy as 83.33% having precision as 80% and recall as 100%.

Introduction

- SpaceX has gained worldwide attention for a series of historic milestones. It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars wheras other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. This dataset includes a record for each payload carried during a SpaceX mission into outer space.

- We would like to find out the predict mission whether it is successful or failed

Methodology



- Data collection methodology:
  - From CSV file through Web
  - From HTML web page through Web Scraping
  - From JSON web file through JSON API
- Perform data wrangling
  - Added new columns to set training labels of the dataset
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build different models using different algorithms, tune, evaluate classification of models.

Methodology

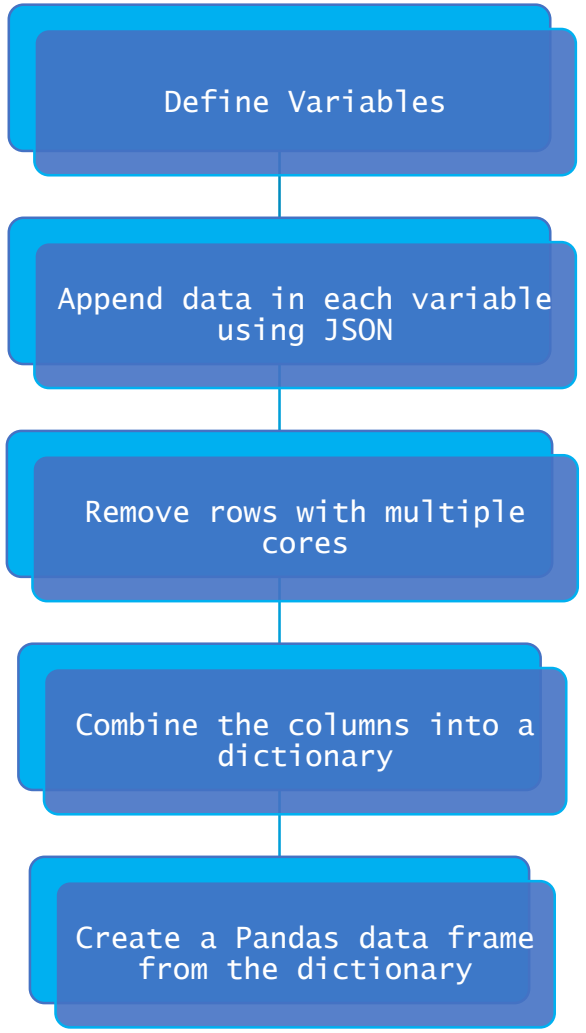
Data collection

- Data sets were collected in the following ways
  - From CSV file through web
  - From HTML web page through Web Scraping
  - From JSON web file through JSON API
- Data collection process is shown using flowcharts in the next section

Data collection – SpaceX API

Solution of the GitHub URL:

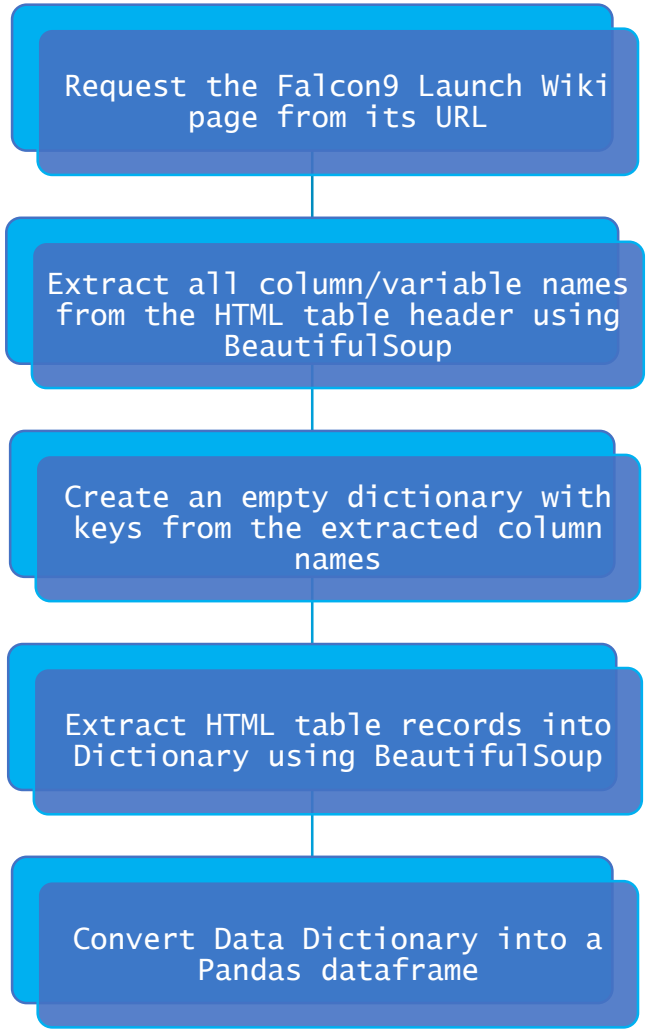
https://github.com/minhajrk/Applied-Data-Science-Capstone



Data collection – Web scraping

Solution of the GitHub URL:

https://github.com/minhajrk/Applied-Data-Science-Capstone



Data wrangling

In the data wrangling stage I have created a new variable to set training labels of the dataset. Here is the steps/flows of the process:

Load Dataset → Create a new list of “bad_outcomes” → Create a new landing class “landing_class” where the element is zero if the corresponding row inis in the set bad outcomes, otherwise, it's one.

My GitHub URL: https://github.com/minhajrk/Applied-Data-Science-Capstone

EDA with data visualization

- We saw different graph to understand the relationship among different variables. Here are some angles:
  - Visualize the relationship between Flight Number and Launch Site
  - Visualize the relationship between Payload and Launch Site
  - Visualize the relationship between Success Rate of each Orbit type
  - Visualize the relationship between FlightNumber and Orbit type
  - Visualize the relationship between Payload and Orbit type
  - Visualize the launch success yearly trend

- My GitHub URL: https://github.com/minhajrk/Applied-Data-Science-Capstone

EDA with SQL

- All launch site names
- Launch site names begin with `CCA`
- Total payload mass
- Average payload mass by F9 v1.1
- First successful ground landing date
- Successful drone ship landing with payload between 4000 and 6000
- Total number of successful and failure mission outcomes
- Boosters carried maximum payload
- 2015 launch records
- Rank success count between 2010-06-04 and 2017-03-20

My GitHub URL: https://github.com/minhajrk/Applied-Data-Science-Capstone

Build an interactive map with Folium

- I have created and added some map objects such as markers, circles, lines, etc. to a folium map

- Markers are used to marked the launching pad, circles are used to marked the launching site and line are used to see the distance between two points.

My GitHub URL: https://github.com/minhajrk/Applied-Data-Science-Capstone

Build a Dashboard with Plotly Dash

- I have added pie chart and scatter plot with Ploty Dash
- To see different success count & rate over different booster version or success over different payload dynamically it is suitable to use Plotly Dash whereas in normal graph it needs to plot multiple times.

My GitHub URL: https://github.com/minhajrk/Applied-Data-Science-Capstone

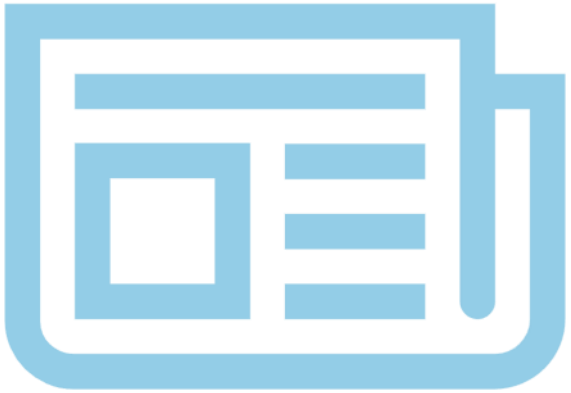
Predictive analysis (Classification)

- Prepared four types of model with best parameters evaluated and found Tree as the best performing classification model among all.

- Model Development Steps:
  1. Data Collection
  2. Exploratory Data Analysis (EDA)
  3. Data Cleansing & Imputation
  4. Feature Selection
  5. Training & Testing Set Preparation
  6. Training Model
  7. Testing & Selecting Best Model

My GitHub URL: https://github.com/minhajrk/Applied-Data-Science-Capstone

Results



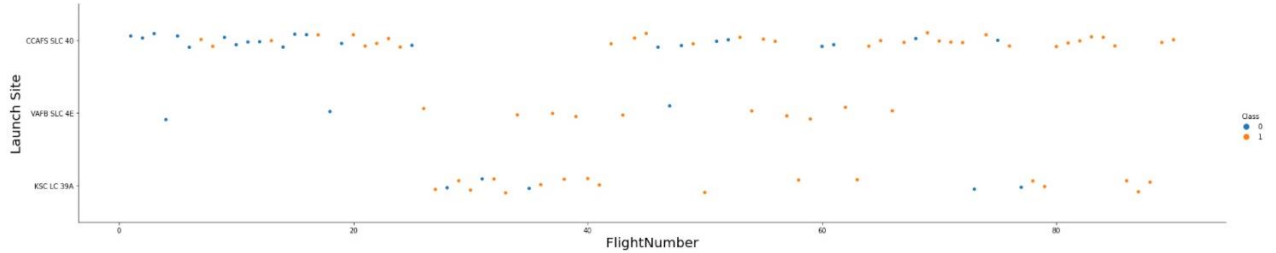
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

EDA with Visualization

Flight Number vs. Launch Site

- To plot a scatter point chart, I have used seaborn catplot function for this purpose.
- I have assigned FlightNumber in the X axis and LaunchSite in the Y axis.
- It seems that as the flight number increases, the first stage is more likely to land successfully.

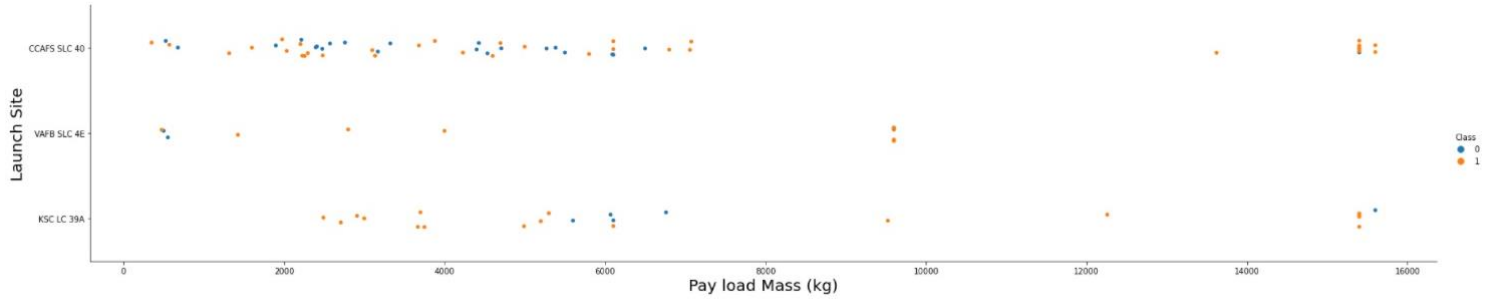
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(x='FlightNumber', y='LaunchSite', data=df, kind='strip',hue='Class', aspect = 5);
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Payload vs. Launch Site

- To plot a scatter point chart, I have used seaborn catplot function for this purpose.
- I have assigned PayloadMass in the X axis and LaunchSite in the Y axis.
- It seems the more massive the payload, the less likely the first stage will return.

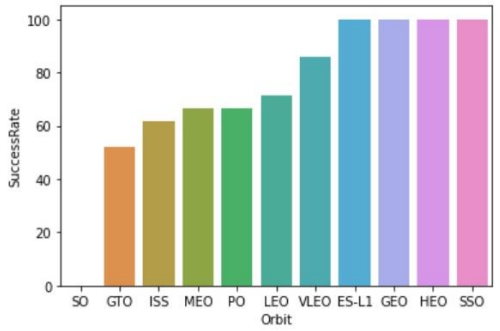
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site,
#and hue to be the class value
sns.catplot(x='PayloadMass', y='LaunchSite', data=df, kind='strip',hue='Class', aspect = 5);
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Success rate vs. Orbit type

- To plot a barchart, I have used seaborn barplot function for this purpose.
- I have assigned Orbit in the X axis and Mean Success Rate in the Y axis.
- It seems ES-L1, GEO, HEO, SEO has the highest success rate GTO has the lowest success rate.

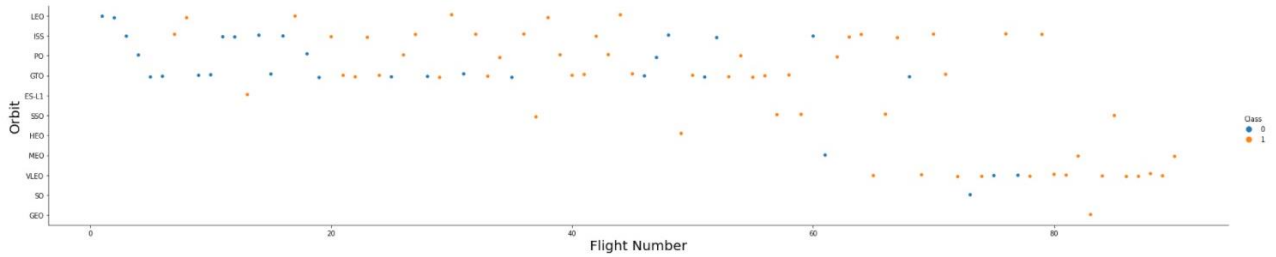
```
# HINT use groupby method on Orbit column and get the mean of Class column
df_success=df.groupby('Orbit')['Class'].agg(Success='sum',Total='count').reset_index()
df_success['SuccessRate']=round(df_success['Success']*100.0/df_success['Total'],2)
df_success=df_success.sort_values(by='SuccessRate', ascending=True)
ax = sns.barplot(x="Orbit", y="SuccessRate", data=df_success)
```



Flight Number vs. Orbit type

- To plot a scatter point chart, I have used seaborn catplot function for this purpose.
- I have assigned FlightNumber in the X axis and Orbit in the Y axis.
- It seems that flights are shifting in the VLEO orbit over time.

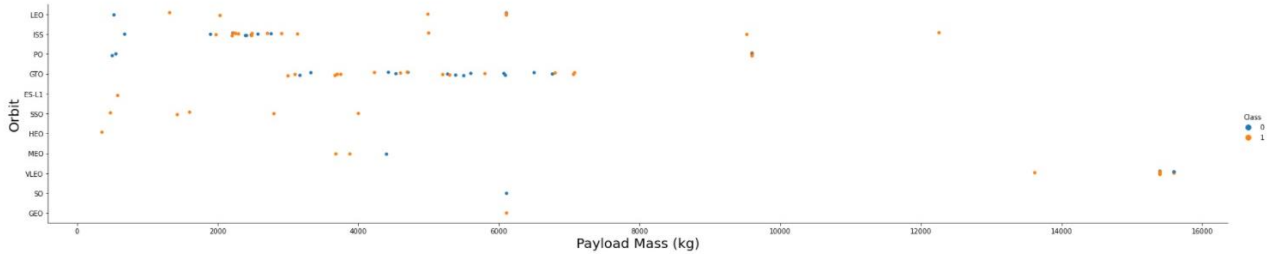
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="FlightNumber", y="Orbit", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Payload vs. Orbit type

- To plot a scatter point chart, I have used seaborn catplot function for this purpose.
- I have assigned PayloadMass in the X axis and Orbit in the Y axis.
- It seems that flights going to VLEO orbit has the highest payload mass.

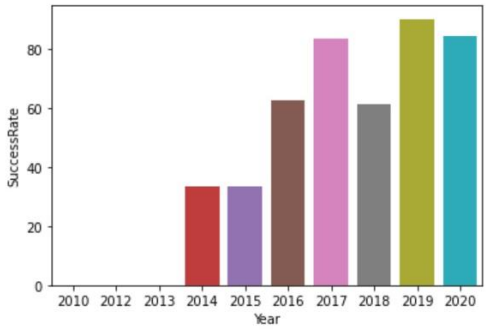
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="PayloadMass", y="Orbit", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch success yearly trend

- To plot a barchart, I have used seaborn barplot function for this purpose.
- I have assigned launch year in the X axis and Mean Success Rate in the Y axis.
- It seems over the year success rate is increasing.

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
y=pd.DataFrame(Extract_year(df['Date']),columns =['Year'])
s=df['Class']
tdf=pd.concat([y, s], axis=1)
df_yearly_success=tdf.groupby('Year')['Class'].agg(Success='sum',Total='count').reset_index()
df_yearly_success['SuccessRate']=round(df_yearly_success['Success']*100.0/df_yearly_success['Total'],2)
df_yearly_success=df_yearly_success.sort_values(by='Year', ascending=True)
ax = sns.barplot(x="Year", y="SuccessRate", data=df_yearly_success)
```



EDA with SQL

All launch site names

```
%sql SELECT DISTINCT launch_site FROM spacextbl;
```

```
 * ibm_db_sa://shv88164:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

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

- I found 5 unique launch site names from the above query. However, it looks there are 4 launch sites in the data (i.e. CCAFS SLC-40 & CCAFSSLC-40 looks same).
- In the above query I used **DISTINCT** keyword. Same result can be found using **GROUP BY** statement as well.

Launch site names begin with `CCA`

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

- Above is the top five result where launch site names begin with ‘CCA’.
- There are 101 records with the above conditions.

Total payload mass

```
%sql SELECT SUM(payload_mass__kg_) total_payload_mass FROM spacextbl WHERE customer LIKE 'NASA%';
```

```
 * ibm_db_sa://shv88164:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

total_payload_mass
99980

- I found total payload mass as 99,980 Kg carried by boosters from NASA.
- Since I have to found out sum of all payload by NASA, we used ‘NASA%’ in the **LIKE** statement.

Average payload mass by F9 v1.1

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

```
 * ibm_db_sa://shv88164:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

avg_payload_mass
2534

- I found average payload mass as **2,534 Kg** carried by boosters version F9 v1.1.
- Since I have to found out average payload by F9 v1.1 booster version, we used ‘F9 v1.1%’ in the **LIKE** statement and used built-in SQL **AVG()** function to found out average payload mass.

First successful ground landing date

```
%sql SELECT MIN(DATE) first_succesful_landing FROM spacextbl WHERE landing__outcome='Success (ground pad)';
```

```
 * ibm_db_sa://shv88164:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

first_succesful_landing
2015-12-22

- I found first successful ground landing date as ‘2015-12-22’.
- To get first successful ground landing, I used built-in SQL **MIN()** function and used condition as **landing__outcome = 'Success (ground pad)'** in the **WHERE** clause.

Successful drone ship landing with payload between 4000 and 6000

```
%%sql
SELECT booster_version
FROM spacextbl
WHERE landing__outcome='Success (drone ship)' AND payload_mass__kg_ BETWEEN 4000 AND 6000
```

```
 * ibm_db_sa://shv88164:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

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

- We found 4 successful drone ship landing with payload between 4000.
- To get the above result we used two conditions in the WHERE clause by AND statement. To apply range value condition we used BETWEEN … AND … statement.
- To use multiple line of SQL statement we used %%sql.

Total number of successful and failure mission outcomes

```
%%sql
SELECT
    SUM(CASE WHEN mission_outcome LIKE 'Success%' THEN 1 ELSE 0 END) Successful_Mission_Count,
    SUM(CASE WHEN mission_outcome LIKE 'Failure%' THEN 1 ELSE 0 END) Failure_Mission_Count
FROM
    spacextbl
```

```
 * ibm_db_sa://shv88164:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

successful_mission_count	failure_mission_count
100	1

- I found 100 successful missions and only 1 failure mission.
- To get the above result I used **CASE WHEN … THEN … ELSE … END** statement at the first stage and then used **SUM()** to calculate successful and failure mission outcomes.
- To use multiple line of SQL statement we used %%sql.

Boosters carried maximum payload

```
%%sql
SELECT booster_version
FROM spacextbl
WHERE payload_mass__kg_ IN(SELECT MAX(payload_mass__kg_) max_payload FROM spacextbl)
```

```
 * ibm_db_sa://shv88164:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
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

- I found 12 boosters and list them with the above query.
- To get the above result I used a sub query to find maximum payload and then I used in the **IN** clause of **WHERE** statement.
- To use multiple line of SQL statement we used %%sql.

2015 launch records

```
%%sql
SELECT MONTHNAME(DATE) MONTH_NAME,landing__outcome,booster_version,launch_site
FROM spacextbl
WHERE landing__outcome='Failure (drone ship)' AND YEAR(DATE)=2015
```

```
 * ibm_db_sa://shv88164:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

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

- I found 2 records and list month name, landing outcome, booster version, launch site with the above query.
- To get the above records I used two conditions in the **WHERE** statement.
- To get month name we used DB2 SQL built-in function **MONTHNAME()** and to apply year condition in the WHERE statement we used DB2 SQL built-in **YEAR()** function.
- To use multiple line of SQL statement we used %%sql.

Rank success count between 2010-06-04 and 2017-03-20

```
%%sql
SELECT DATE,count(1) Success_Landing_Count
FROM spacextbl
WHERE landing__outcome LIKE 'Success%' AND DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY DATE
ORDER BY 2 DESC
```

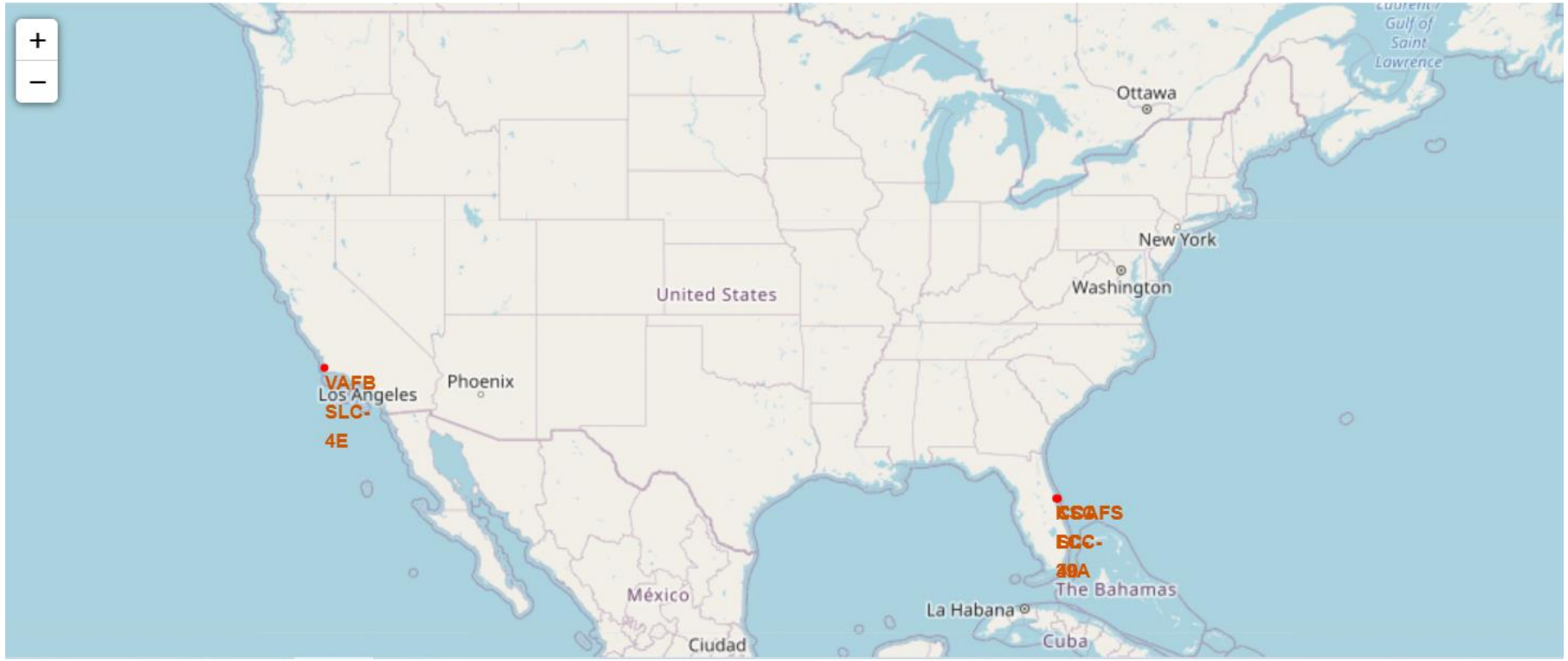
```
 * ibm_db_sa://shv88164:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

DATE	success_landing_count
2015-12-22	1
2016-04-08	1
2016-05-06	1
2016-05-27	1
2016-07-18	1
2016-08-14	1
2017-01-14	1
2017-02-19	1

- I found 8 records and list month name, landing outcome, booster version, launch site with the above query.
- To get the above records I used two conditions in the **WHERE** statement.
- To get month name we used DB2 SQL built-in function **MONTHNAME()** and to apply year condition in the WHERE statement we used DB2 SQL built-in **YEAR()** function.
- To use multiple line of SQL statement we used %%sql.

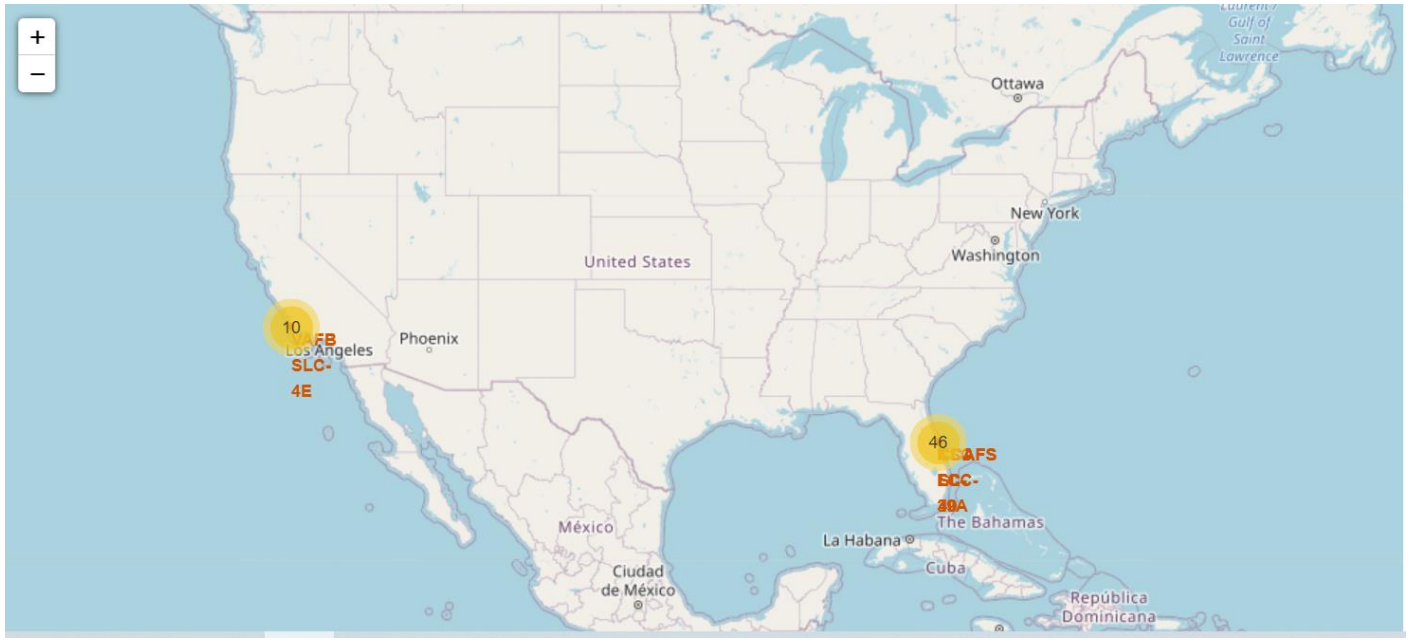
Interactive map with Folium

Mark all launch sites on a map



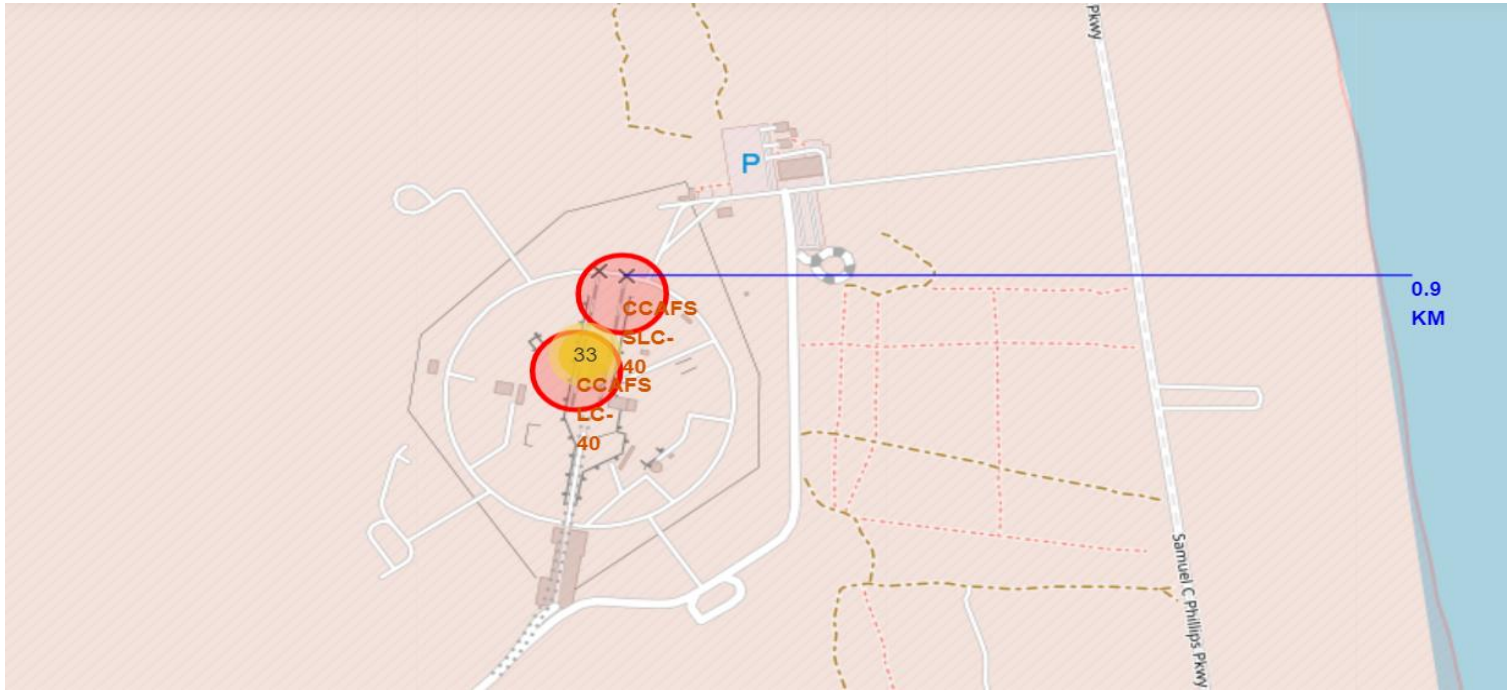
- At first I have separated Launch Site', 'Lat', 'Long' from the loaded dataset. Then uniquely identified these value by groupby function.
- Then initialize the folium map.
- Then I have created circle & marker objects of all launch sites and added to the map.
- Finally display the map to see the effect.

Mark the success/failed launches for each site on the map



- Firstly I added a new column "**marker_color**" (Red for failed and green for successed) in the spacex_df.
- Then added the Marker cluster to the site map for each row in spacex_df data frame.
- Then I have created a Marker object with its coordinate and customize the Marker's icon property to indicate if this launch was successed or failed.
- Finally display the map to see the effect.

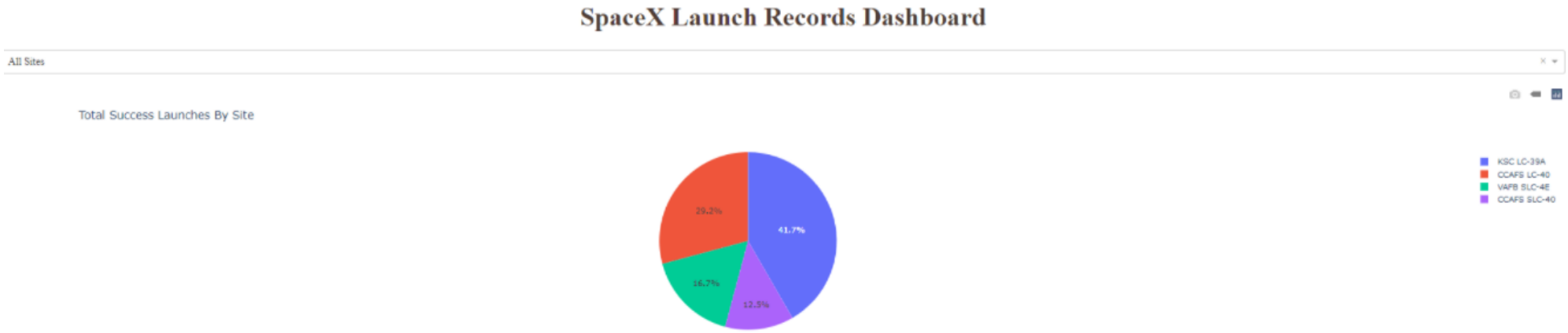
Calculate the distances between a launch site to its proximities



- I have added a MousePosition on the map to get coordinate for a mouse over a point on the map.
- Zoomimg into a launch site, exploring its proximity and moving my mouse to these points and mark down their coordinates (shown on the top-left) in order to note down the points.
- To get distance I have used a given function “calculate_distance”.
- Then I have used PolyLine to draw a line between two points and I have used Marker to display the distance between two points.
- Finally displayed the map to see the effect

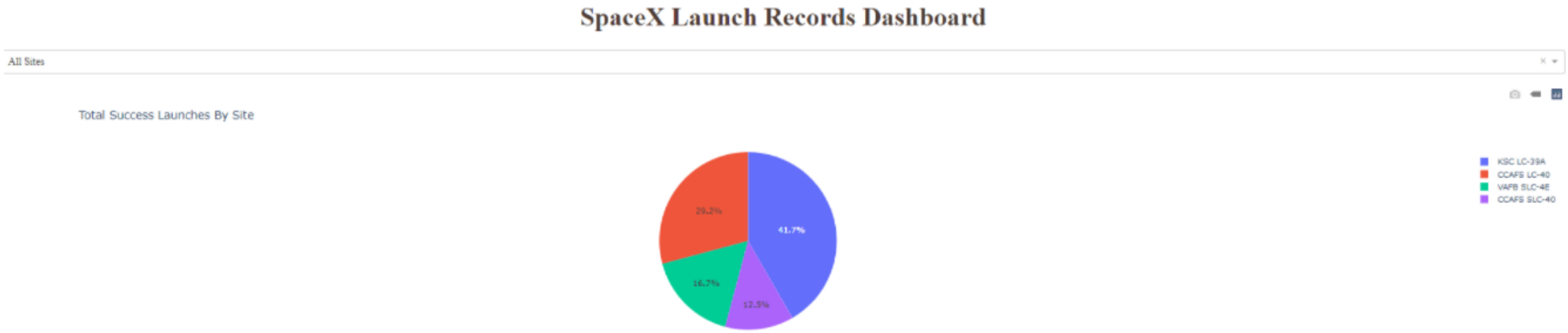
Build a Dashboard with Plotly Dash

Successful Launch by Launch Sites



- Since the launch sites were only 4, so pie chart was suitable for the presentation.
- From the above pie chart, it seems that KSC LC-39A has the highest success c launch site followed by CCAFS SLC-40.

Successful Launch Rate by Launch Sites



- Since the launch sites were only 4, so pie chart was suitable for the presentation.
- From the above pie chart, it seems that KSC LC-39A has the highest success launch rate followed by CCAFS SLC-40.

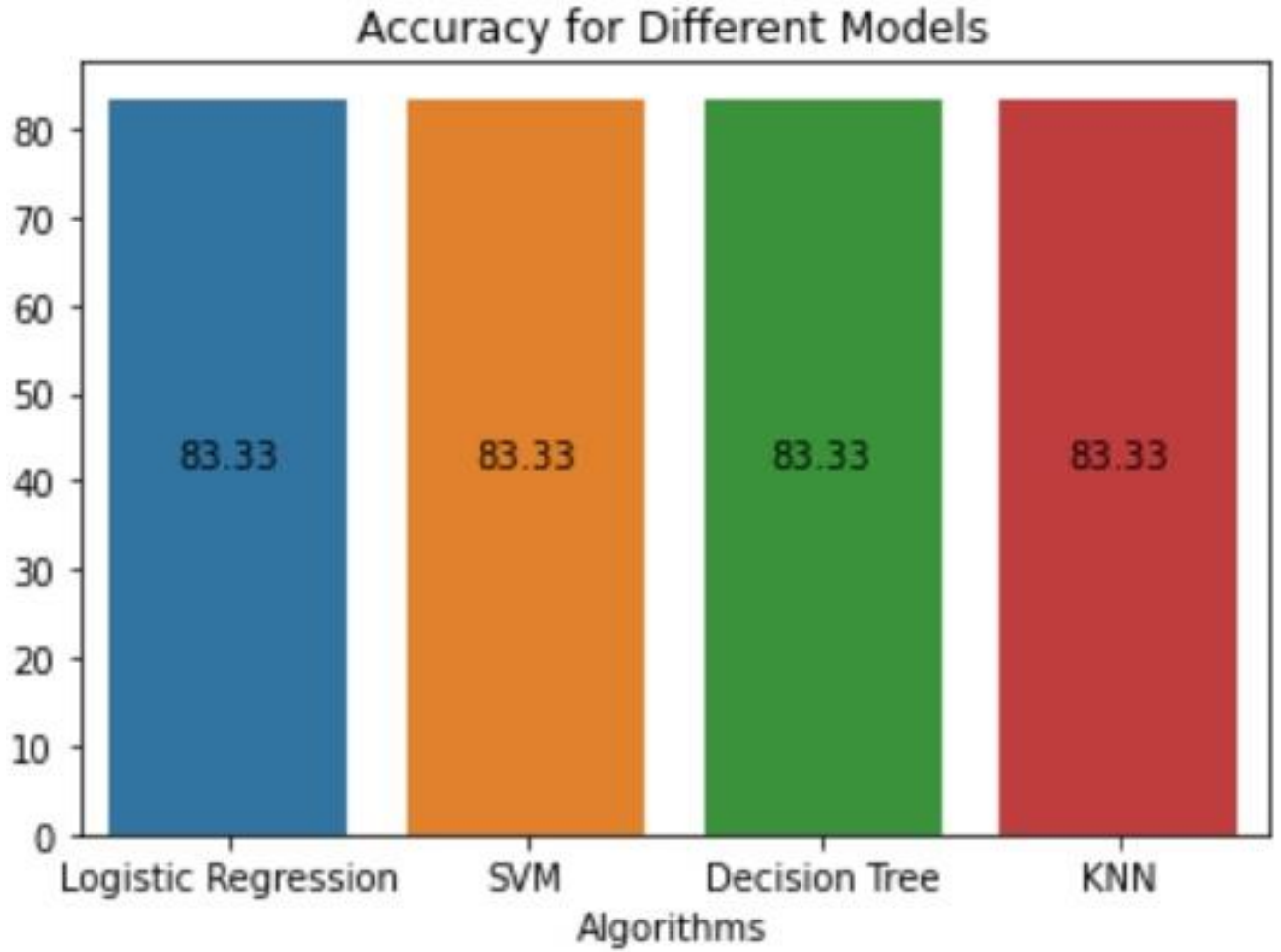
Correlation Between Payload and Success for all Sites

- Above figure shows the correlation between payload and success for all Sites.
- Booster version v1.1 has the highest failure rate followed by booster version v1.0 in from the lower payload to higher payload.

Predictive analysis (Classification)

Classification Accuracy

When we applied our built models with test data set, it seems that all models shows same classification accuracy of 83.33%. So, we can select any of the models.



Confusion Matrix

Some important Confusion Matrix Parameters:

Accuracy = (TN+TP)/(TN+TP+FP+FN)

Precision = TP/(TP+FP)

Recall = TP/(TP+FN)

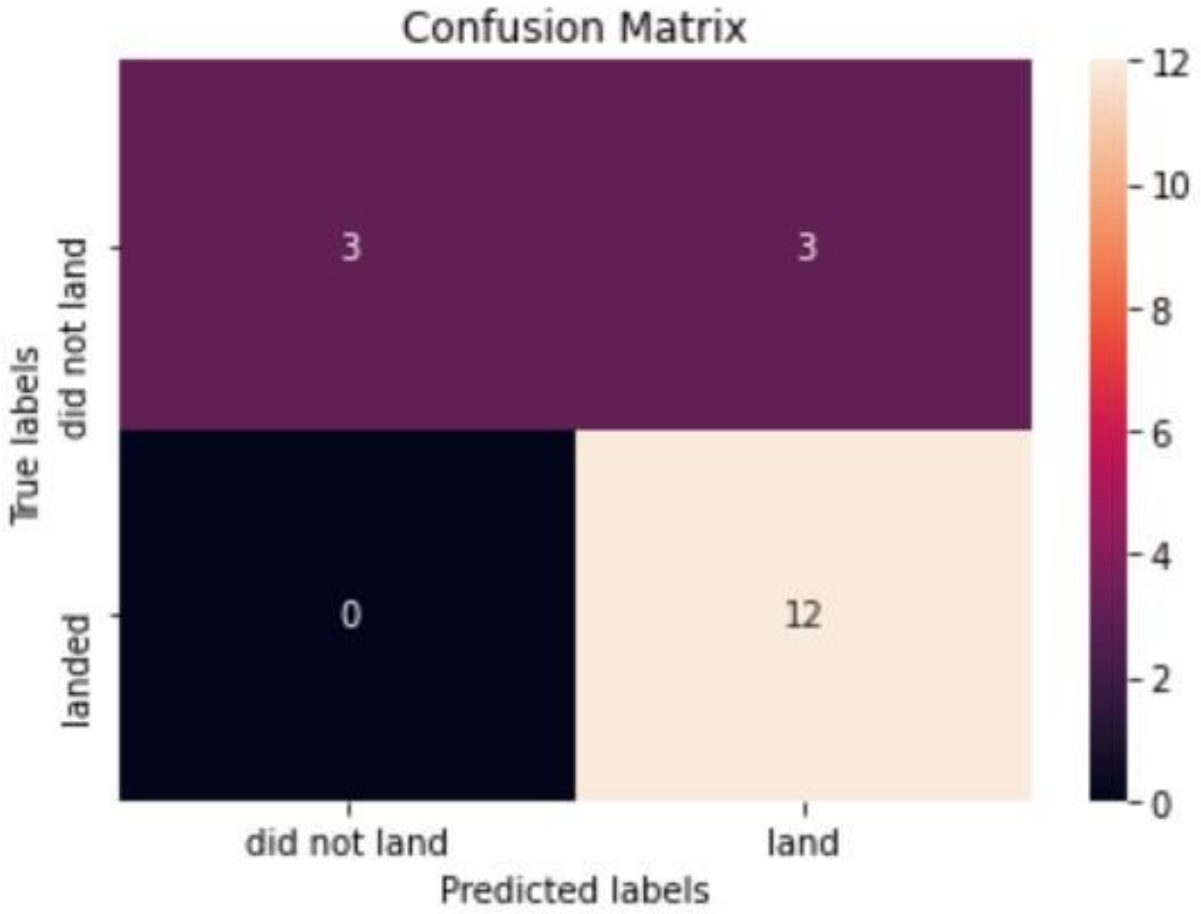
So, putting value we can get:

Accuracy % = 83.33 %

Precision % = 80 %

Recall % = 100 %

According to the above figure it seems that my model successfully identified all failure landed case but some successful landed cases are identified as failed case.



CONCLUSION

- According to data it seems that SpaceX increased their capacity and success rate over time.
- All models showed same accuracy (83.33%) and f1 score (88.89%).
- Models predicted all failure landing as failure and some success landing as failure.
- For the investors, recall rate is very much important and here we have seen that rate was 100%. So, we can say that investors can plan their investment considering this parameter.

APPENDIX

- Final dataset link:
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv
  - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_3.csv
- Program Used
  - Python on cloud as well as on local machine
  - IBM DB2 on cloud
- Packages Used under Python:
  - Pandas
  - Numpy
  - Seaborn
  - Sklearn
  - Request
  - BeautifulSoup
  - Re
  - Plotly