IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

ROHIT CHANDRAN V

1ST OCTOBER 2024

OUTLINE

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

EXECUTIVE SUMMARY

SUMMARY OF METHODOLOGIES

I. **Data Collection and Preprocessing**: Ensured high-quality data through cleaning, normalization, and encoding.

II. **Exploratory Data Analysis**: analyzed and visualized data to understand its key characteristics, uncover patterns, and identify relationships between variables.

III. **Feature Engineering**: Created derived features to enhance predictive power.

IV. **Visualizations and Insights**: Provided clear visualizations and insights into key factors influencing landing success and cost implications.

V. **Predictive Modeling**: Used a Random Forest Classifier to predict landing success with high accuracy.

SUMMARY OF ALL RESULTS

- Overall result was satisfactory. Gathered insights based on patterns found during EDA & Visualization. SVM Model predicted correct data & solved questions raised during discussion. Cost was determined.

INTRODUCTION

- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX 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 report presents a detailed analytic report based on information gathered about SpaceX Rocket launches.
- Problems that needs to be answered.
  - i. Predict whether falcon 9's first stage will land successfully.
  - ii. To determine whether the first stage will be re used or not & thereby determine overall cost.

Section 1

Methodology

METHODOLOGY

- ➢ DATA COLLECTION: Gathered data from REST API & Wiki page.
- ➢ DATA WRANGLING:
  - one Hot Encoding data fields for Machine Learning & data cleaning of null values and irrelevant columns.
- ➢ EXPLORATORY DATA ANALYSIS (EDA) USING VISUALIZATION AND SQL.
- ➢ FEATURE ENGINEERING
- ➢ INTERACTIVE VISUAL ANALYTICS USING FOLIUM AND PLOTLY DASH.
- ➢ PREDICTIVE ANALYSIS USING CLASSIFICATION MODELS.
  - LR, KNN, SVM, DT models have been built and evaluated for the best classifier.

DATA COLLECTION

- Data was collected from two sources:
  a) SpaceX REST API endpoints (api.spacexdata.com/v4/launches/past)
  b) https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches via web scrapping.

- Data was collected using Python libraries & packages.

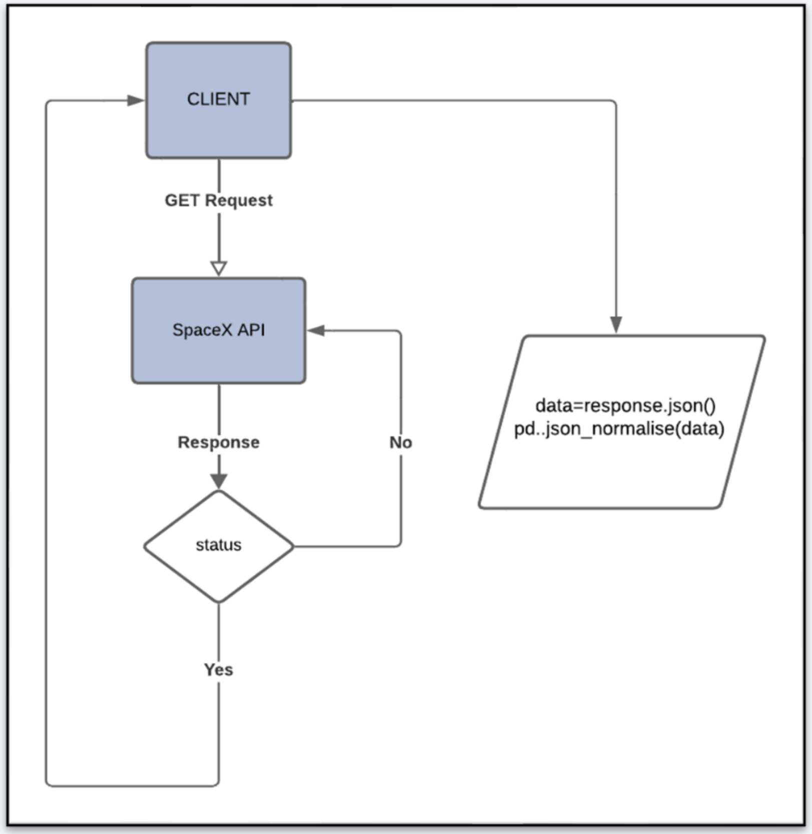
- Variables used

  Key variables included Flight Number, Date, Booster Version, Payload Mass, Orbit, Launch Site, Outcome, Flights, Grid Fins, Reused, Legs, Landing Pad, Block, Reused Count, Serial, Longitude of launch site , Latitude of launch site.

- Data Quality : Lot of the data are IDs. for e.g., the rocket column has no information about the rocket just an identification number. Used the API again to get information about the launches using the IDs given for each launch

DATA COLLECTION – SPACEX API

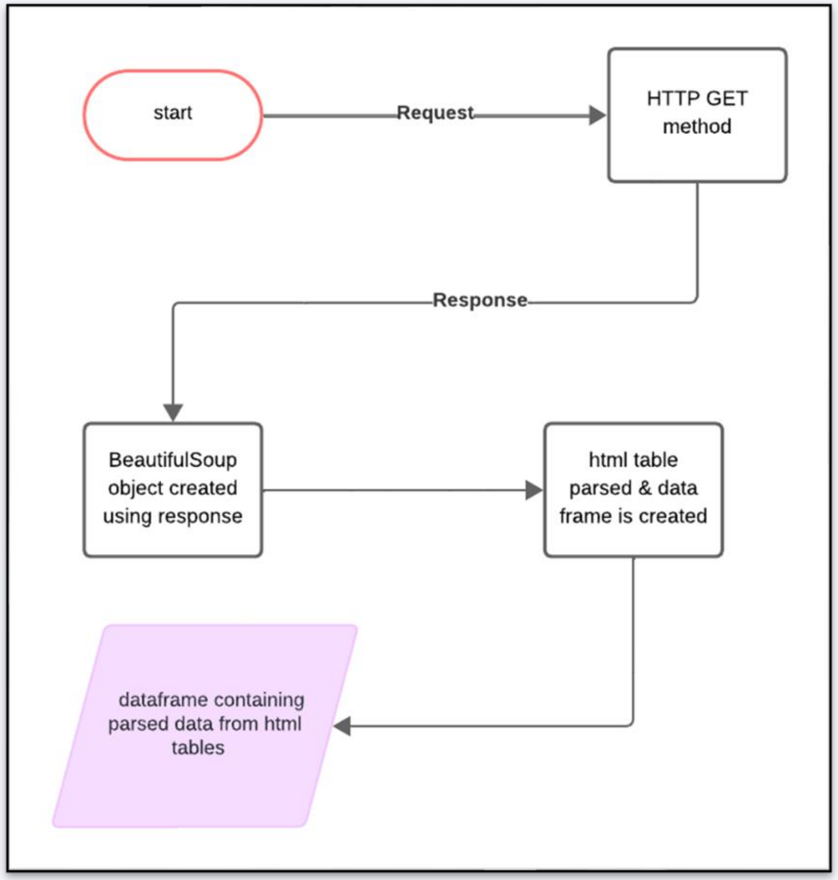
- Data was collected by requesting launch data from SpaceX API using endpoint: api.spacexdata.com/v4/launches/past
- Data shall be received as Json objects.
- Wrangling data using API: Some of the column contain only identification number & hence need target other endpoints for specific data using Identification number.
- Sampling Data
- Dealing with Nulls
- Data is normalized & saved as CSV file for Data wangling.



https://github.com/rohitchandran97/Winning-Space-Race-using-Data-Science/blob/d6e58bdddb5430dfa8ccd1fa5b5fed6822816152/jupyter-labs-spacex-data-collection-api.ipynb

DATA COLLECTION - SCRAPING

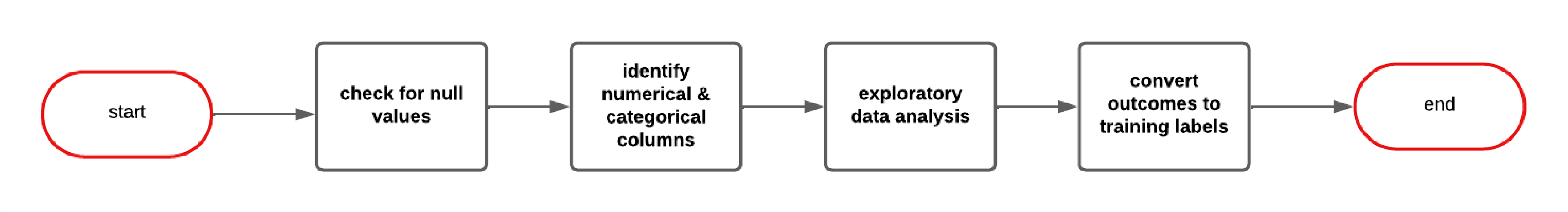
- Requested the Falcon9 Launch Wiki page from its URL.
- Used BeautifulSoup to create a BeautifulSoup object from a response text content.
- Extract all column/variable names from the HTML table header.
- Html table values are parsed into an empty dictionary.
- Created a data frame using the parsed launch Html tables
- DataFrame is saved to a csv file.



https://github.com/rohitchandran97/Winning-Space-Race-using-Data-Science/blob/d6e58bdddb5430dfa8ccd1fa5b5fed6822816152/jupyter-labs-webscraping.ipynb

DATA WRANGLING

1. Check for null values.
2. Identify which columns are numerical & categorical.
3. Do EDA to find pattern in the data & determine what would be the label for training supervised models.
4. convert those outcomes into Training Labels with 1 means the booster successfully landed & 0 means it was unsuccessful.
5. Newly created dataset is then saved as separate csv file.



https://github.com/rohitchandran97/Winning-Space-Race-using-Data-Science/blob/d6e58bdddb5430dfa8ccd1fa5b5fed6822816152/labs-jupyter-spacex-Data%20wrangling-v2.ipynb

EDA WITH DATA VISUALIZATION

i. Created scatter plots to see how two variables affects launch outcome.
   - Payload mass vs Flight no
   - Flight no vs launch site
   - Payload vs launch site
   - Orbit type vs flight no
   - Orbit type vs Payload mass

ii. Bar chart to visualize success rate for each orbit type.

iii. Line chart to visualize success rate over the years.

https://github.com/rohitchandran97/Winning-Space-Race-using-Data-Science/blob/d6e58bdddb5430dfa8ccd1fa5b5fed6822816152/Data_ExploratoryDataAnalysis.ipynb

FEATURES ENGINEERING

After evaluating the patterns & insights gathered, certain features are moved to a new *DataFrame*.

Function *OneHotEncoder* is implemented on columns *Orbit ,LaunchSite , LandingPad* and *Serial.*

Assigned the value to the variable *features_one_hot.*

Casted the entire *DataFrame* to variable type *float64*.

EDA WITH SQL

- Displayed the names of the unique launch sites in the space mission.
- Displayed 5 records where launch sites begin with the string 'CCA'.
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display average payload mass carried by booster version F9 v1.1.
- Listed the date when the first successful landing outcome in ground pad was achieved.
- Listed the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- List the total number of successful and failure mission outcomes.

https://github.com/rohitchandran97/Winning-Space-Race-using-Data-Science/blob/d6e58bdddb5430dfa8ccd1fa5b5fed6822816152/jupyter-labs-eda-sql-coursera_sqllite.ipynb

- List the names of the booster versions which have carried the maximum payload mass.
- List the records which will display the month names, failure landings in drone ship, booster versions, launch site for the months in year 2015.
- Ranked the count of landing outcomes between the date 2010-06-04 & 2017-03-20, in descending order.

Note: Dataset used includes a record for each payload carried during a SpaceX mission into outer space.

Date	TEXT	"Date" TEXT
Time (UTC)	TEXT	"Time (UTC)" TEXT
Booster_Version	TEXT	"Booster_Version" TEXT
Launch_Site	TEXT	"Launch_Site" TEXT
Payload	TEXT	"Payload" TEXT
PAYLOAD_MASS__KG_	INT	"PAYLOAD_MASS__KG_" INT
Orbit	TEXT	"Orbit" TEXT
Customer	TEXT	"Customer" TEXT
Mission_Outcome	TEXT	"Mission_Outcome" TEXT
Landing_Outcome	TEXT	"Landing_Outcome" TEXT

BUILD AN INTERACTIVE MAP WITH FOLIUM

- Created a folium Map object, with an initial center location to be NASA Johnson Space Center at Houston, Texas.
- Used map object Circle to highlight area around the launch site.
- A marker object is created to add a marker with a launch site name as label, to mark the launch site.
- Created a folium object called *MarkerCluster* using class column values to indicate success & failed launches in respective launch site.
- Added a *MousePosition* object on the map to get coordinates of the curser over a point on the map. As such, while you are exploring the map, you can easily find the coordinates of any points of interests (such as railway).
- Calculated distance from launch site to all points of interests & using *PolyLine* drew lines to indicate distance between each point of interests such as railway, coastline and nearest city to launch site.

https://github.com/rohitchandran97/Winning-Space-Race-using-Data-Science/blob/d6e58bdddb5430dfa8ccd1fa5b5fed6822816152/lab_jupyter_launch_site_location.ipynb

BUILD A DASHBOARD WITH PLOTLY DASH

- Added a Launch Site Drop-down Input Component.
- From drop-down you can choose success rate of a particular launch site or overall success rate.
- Added a callback function to render success-pie-chart based on selected site dropdown.
- Added a Range Slider to Select Payload.
- Added a callback function to render the success-payload-scatter-chart scatter plot.
- Scatter plot displays relationship between payload & success rate for each launch sites.

https://github.com/rohitchandran97/Winning-Space-Race-using-Data-Science/blob/4056a9546e1c6694088e48564af9b6ee3fa3e3b6/spacex_dash_app.py

PREDICTIVE ANALYSIS (CLASSIFICATION)

- Used two sets of datasets which were saved as two different CSV files.
- Data from 1st set was standardized & transformed to variable X.
- Data from class column from 2nd set was assigned to variable Y using NumPy.
- Used the function *train_test_split* to split the data X and Y into training and test data.
- Create a logistic regression object then create a *GridSearchCV* object
- Fitted the object to find the best parameters from the dictionary *parameters*.

https://github.com/rohitchandran97/Winning-Space-Race-using-Data-Science/blob/407047b9cafa25a270347e9c06b6a7dd7552b9c6/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

```python
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

```python
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()

```

```python
# Created a GridSearchCV object with cv = 10
logreg_cv = GridSearchCV(estimator=lr, param_grid=parameters, cv=10)

# Fit the GridSearchCV object to find the best parameters
logreg_cv.fit(X_train, Y_train)
```

```
▸ GridSearchCV
▸ estimator: LogisticRegression
    ▸ LogisticRegression
```

- Displayed the best parameters and the accuracy on the validation data.
- Calculate the accuracy on the test data.
- Created confusion matrix from the above data.
- Repeated these steps using following machine learning algorithms :
  1. SVM
  2. Decision Trees
  3. K-Nearest Neighbors
- Compared all models by comparing accuracies produced by each models.

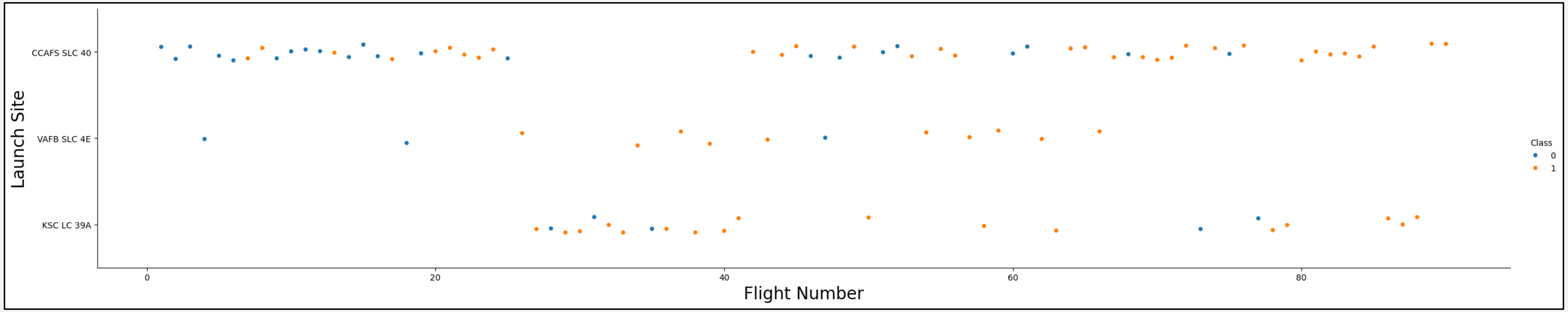
RESULTS

- Post EDA, it was found that certain variables such as *FlightNumber PayloadMass, Orbit, LaunchSite, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount* and *Serial* shall affect the success rate.
- Interactive map displayed geographical patterns about launch sites that includes various POI around launch sites.
- Interactive dashboard provided an interface to render between payload range to understand how it affects success rate of all booster versions based on option selected.
- Best performing Model was found by ranking all models based on their accuracy factor.

Section 2

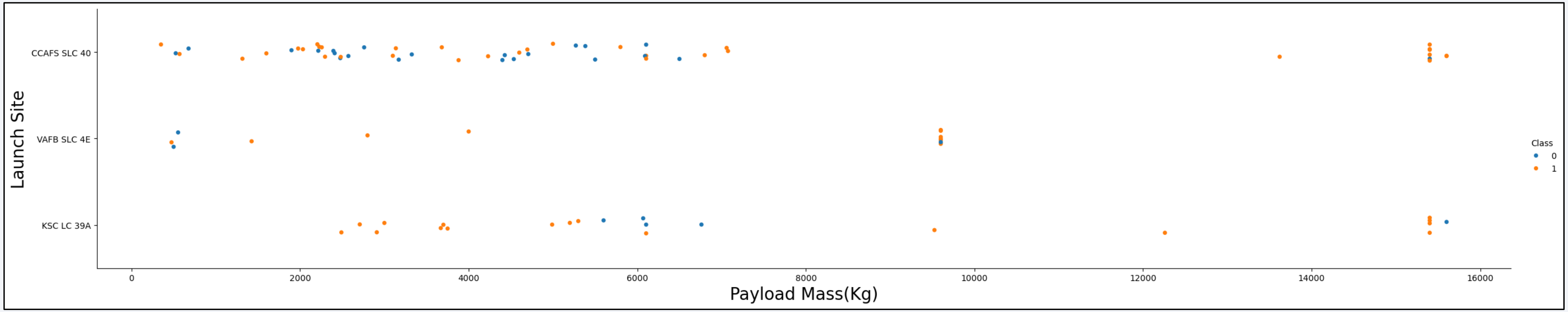
Insights drawn from EDA

FLIGHT NUMBER VS. LAUNCH SITE



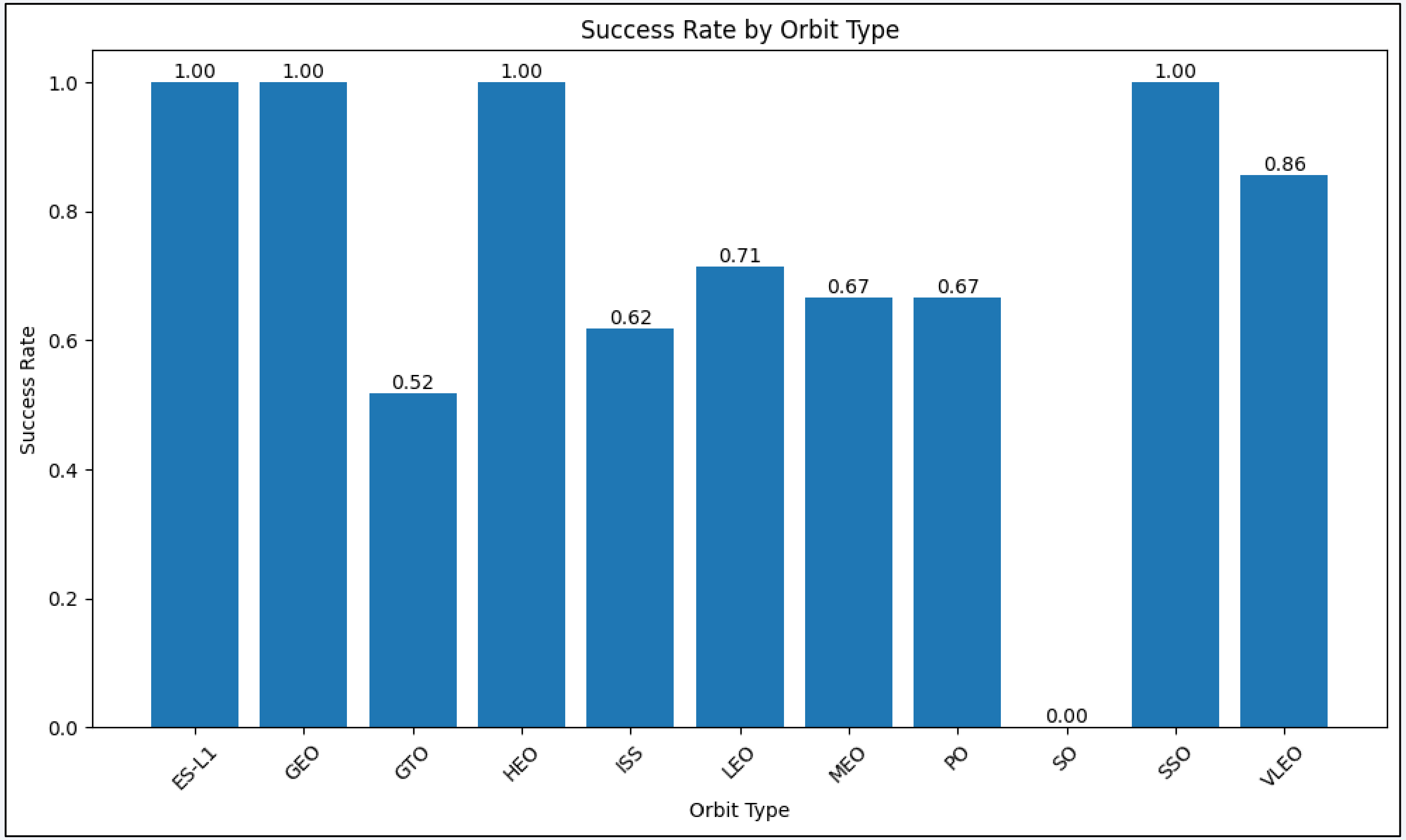
- CCAFS SLC 40 holds more flight no with better outcomes.
- KSC LC 39A and VAFB has better outcome with increased flight number.

PAYLOAD VS. LAUNCH SITE



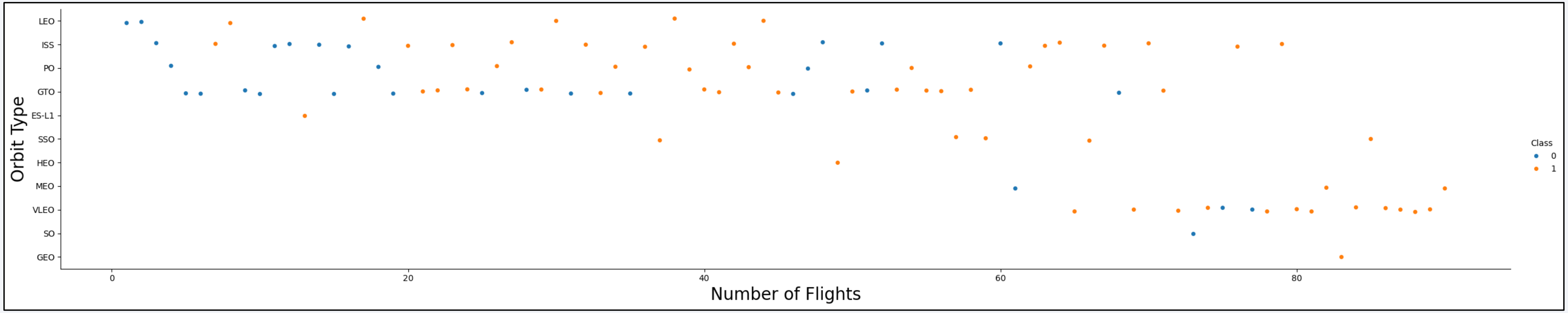
- More successful outcomes with less payload at CCAFS SLC 40.
- Not much launches were made by carrying payload more than 1000.
- No rocket launches were made for payload greater than 1000 at VAFB SLC.

SUCCESS RATE VS. ORBIT TYPE



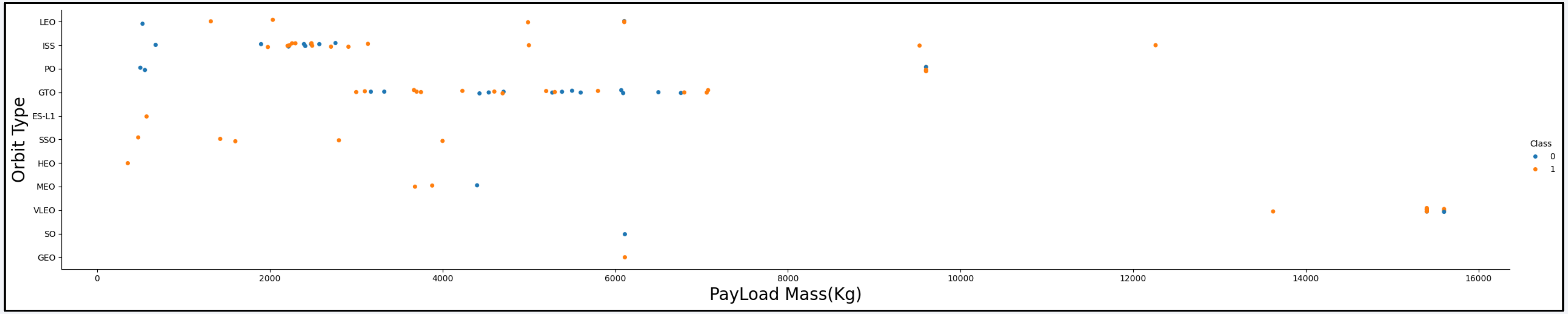
- ES-L1,HEO,SSO has highest success rate.

FLIGHT NUMBER VS. ORBIT TYPE



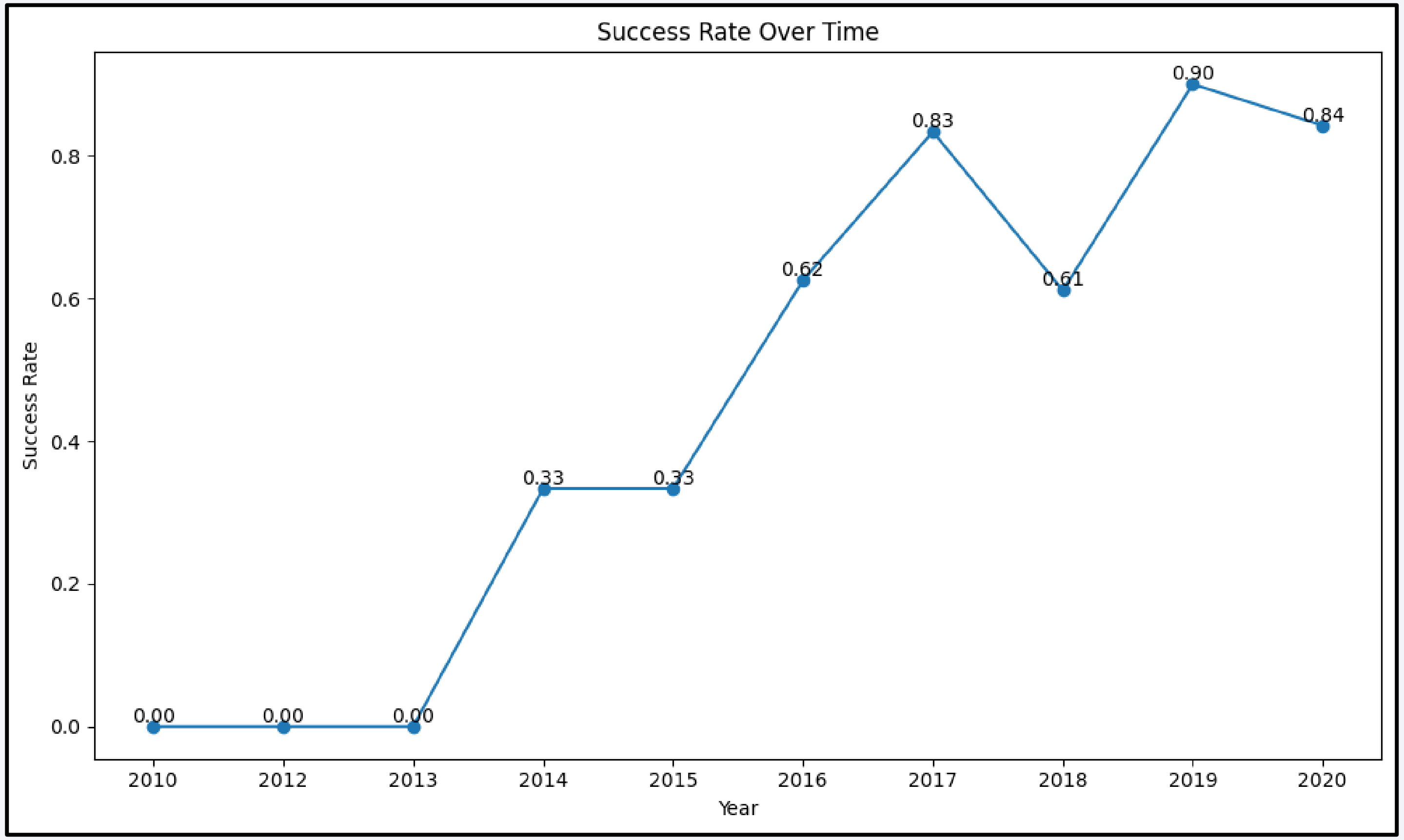
- GTO PO ISS & LEO has made more flights than any other orbit type.
- These orbit types has better success rates for flight number more than 40.
- GTO has better success-failure ratio(14:13).
- LEO don’t have flight number that exceeds 40.

PAYLOAD VS. ORBIT TYPE



- Rockets with Payload greater than 6K don't make unsuccessful landing for any orbit type.
- GEO,HEO,SSO,ES-L1 don't have any unsuccessful outcome.
- GTO orbit type is more associated with rocket launches with Payload ranging between 3000 Kg – 8000kg.

LAUNCH SUCCESS YEARLY TREND



- Sees an upward growth in success rate in rocket launches from 2013 till 2020.
- 2019 saw the highest success rate

All Launch Site Names

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

- Unique launch site was retrieved using **SELECT DISTINCT** statement.
- There are 4 Launch sites.

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

- Used the SELECT statement along with the LIKE operator and the LIMIT clause to retrieve launch sites beginning with “ CCA”.
- There are five records that satisfy this query.

TOTAL PAYLOAD MASS

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)';
```

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

total_payload_mass
45596

- Result was retrieved using query that includes SUM operator for calculating total payload mass and used FROM & WHERE clauses for retrieving Payload mass of boosters launched by NASA.

AVERAGE PAYLOAD MASS BY F9 V1.1

* sqlite:///my_data1.db
Done.

average_payload_mass
2534.6666666666665

Used AVG operator and WHERE and LIKE clauses to retrieve average payload mass for certain versions of booster.

FIRST SUCCESSFUL GROUND LANDING DATE

```
%%sql
SELECT MIN(Date) AS First_Successful_Ground_Pad_Landing_Date
FROM SPACEXTABLE
WHERE Mission_Outcome = 'Success'
AND Landing_Outcome = 'Success (ground pad)';
```

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

First_Successful_Ground_Pad_Landing_Date
2015-12-22

- Used MIN operator, WHERE & AND Keywords to get date of first successful ground landing.

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

- Used greater than & lesser than operators to obtain following result.
- 4 records were obtained for the given query.

```
%%sql
SELECT DISTINCT Booster_Version AS Booster_Name
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ > 4000
AND PAYLOAD_MASS__KG_ < 6000;
```

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

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

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

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

Total_Successful_Missions	Total_Failed_Missions
100	1

- Obtained no of counts of successful & unsuccessful using CASE,LIKE clauses and THEN-ELSE-END statement inside SUM operator.

BOOSTERS THAT CARRIED MAXIMUM PAYLOAD

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

Boosters with maximum payload Mass was obtained using subquery where MAX operator was used to retrieve maximum payload mass value.

12 records were obtained.

2015 LAUNCH RECORDS

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

- Used SUBSTR(Date, 6,2) as month to get the months and SUBSTR(Date,0,5)='2015' for year.
- Lists failed outcomes in drone ship, their booster versions, and launch site in year 2015.

RANK LANDING OUTCOMES BETWEEN 2010-06-04 AND 2017-03-20

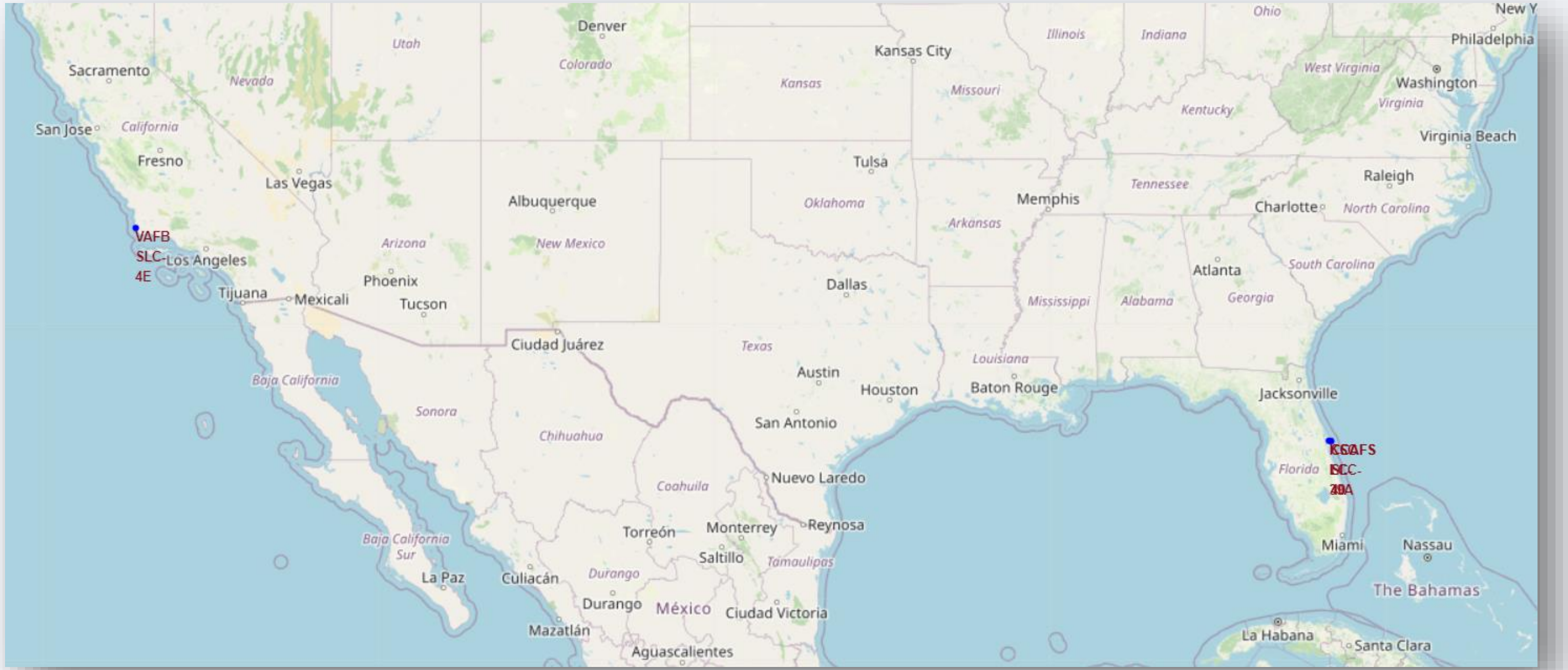
Landing_Outcome	Count_Landing_Outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- Used COUNT operator for counting and used GROUP BY clause to group by landing outcome. Then used ORDER BY clause to rank them in particular order.
- In most cases there were no attempt made to land.

Section 3

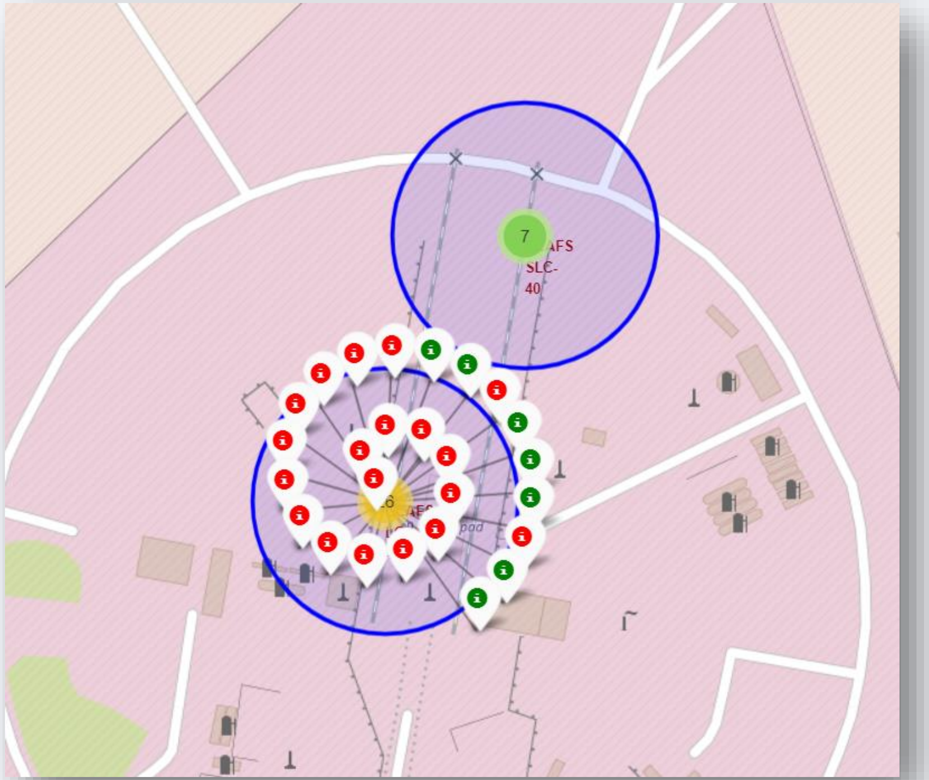
Launch Sites Proximities Analysis

ROCKET LAUNCH SITES



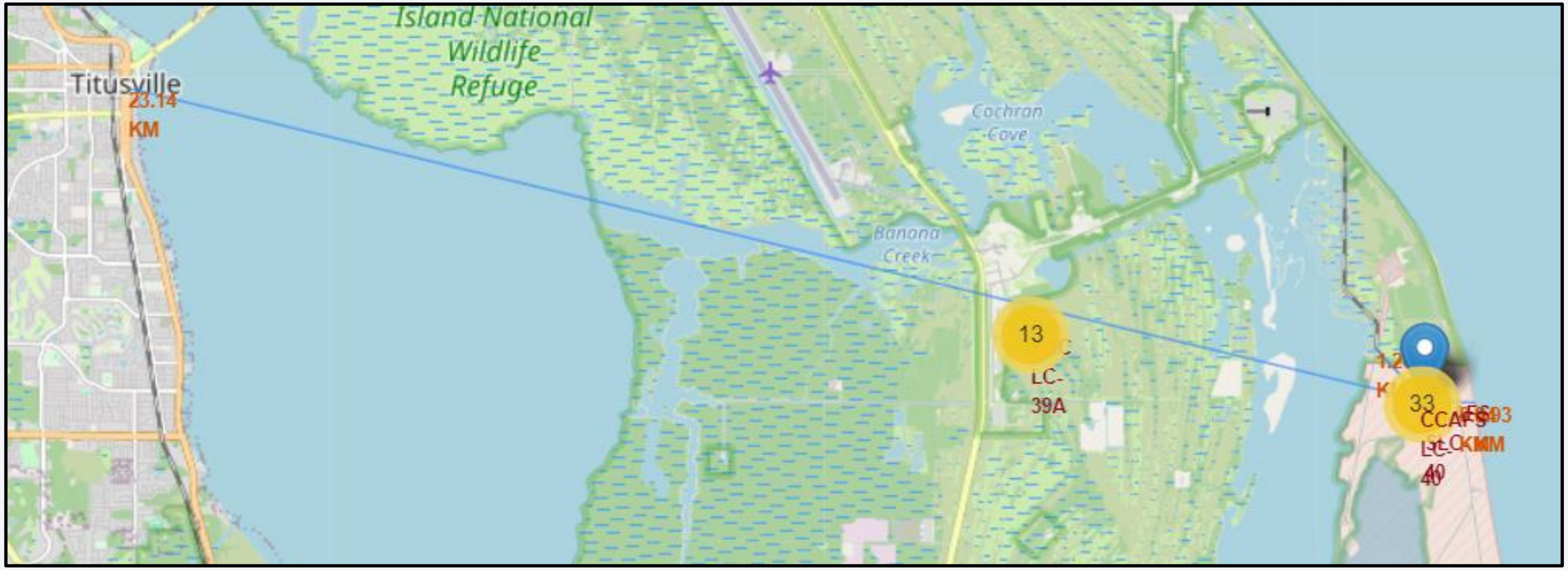
Total four launch sites are being displayed. Three of them are close to each other but one is located on the other side of USA.

LAUNCH OUTCOMES FOR EACH SITE



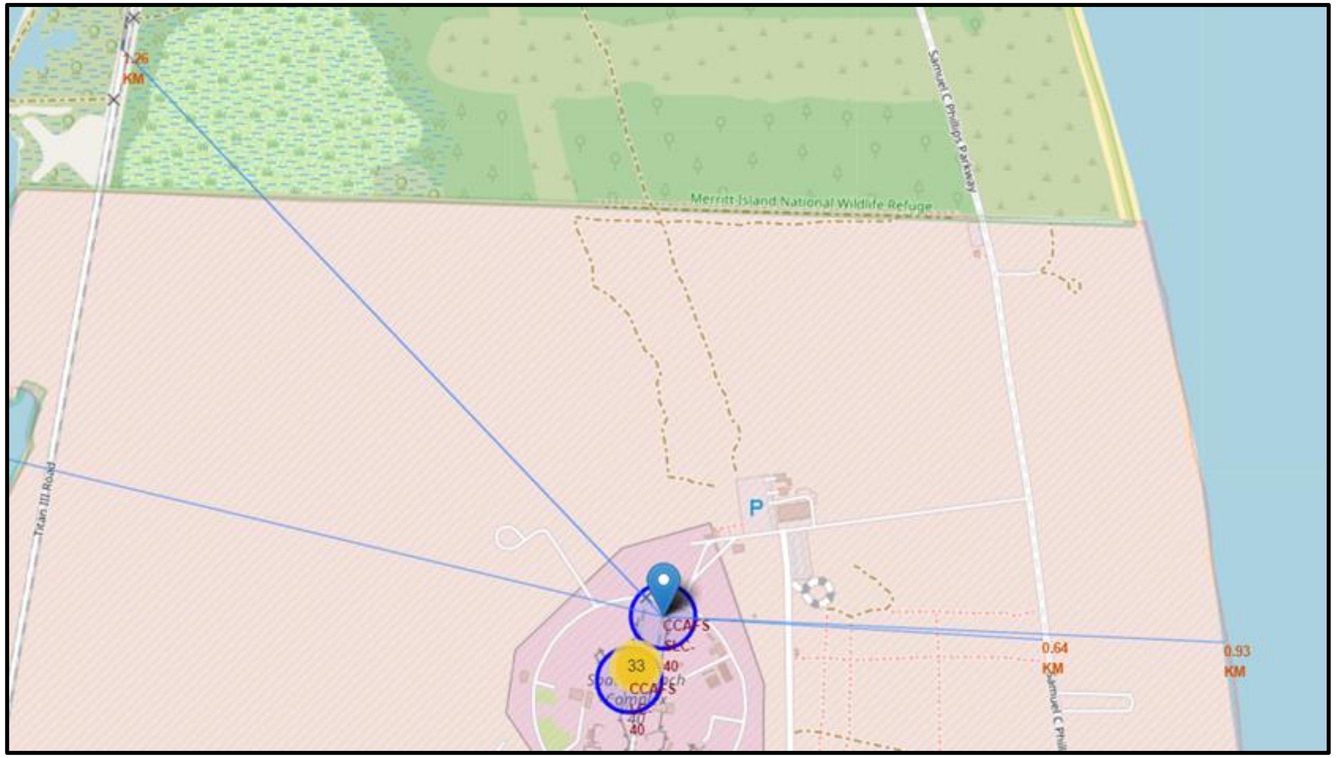
Totally 26 launches were made at CCAFS LC 40 and out of it, 19 were unsuccessful.

PROXIMITIES OF LAUNCH SITES



Line indicating distance between CCAFS SLC 40 to nearest city.

DISTANCE FROM LAUNCH SITE TO NEAREST POI

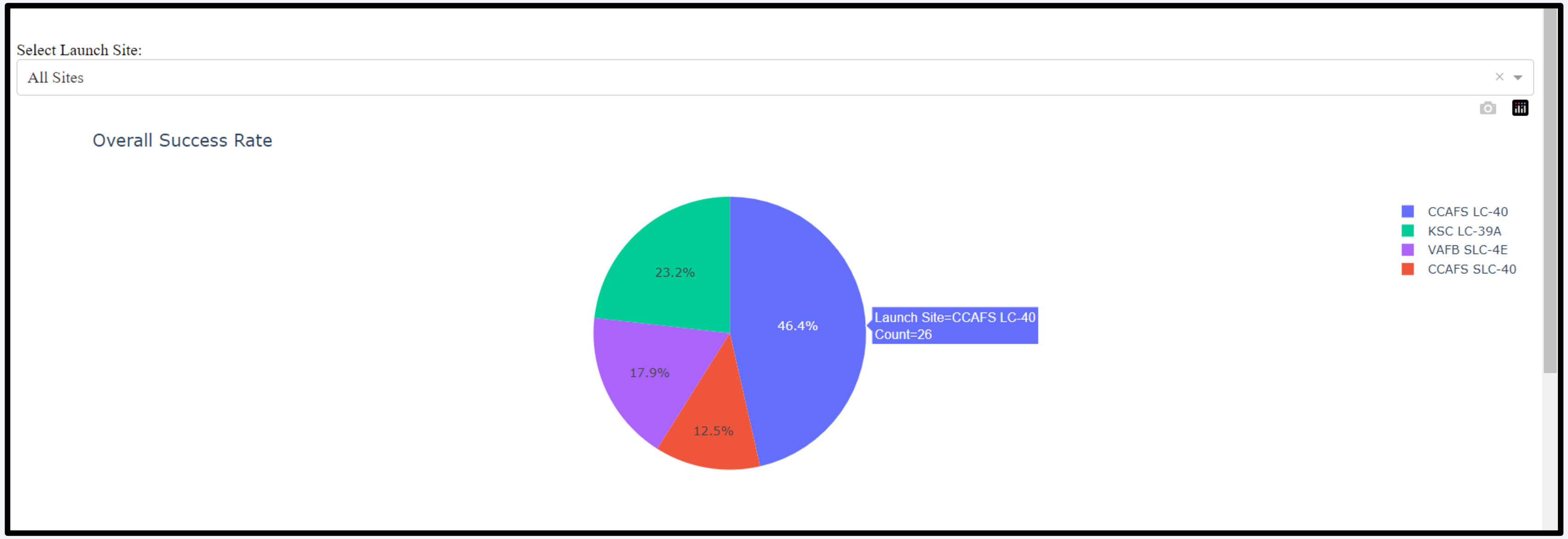


- Coastline : 0.94 km
- Railway : 1.26 km
- Highway : 0.64 km
- City : 23.14 km

Section 4

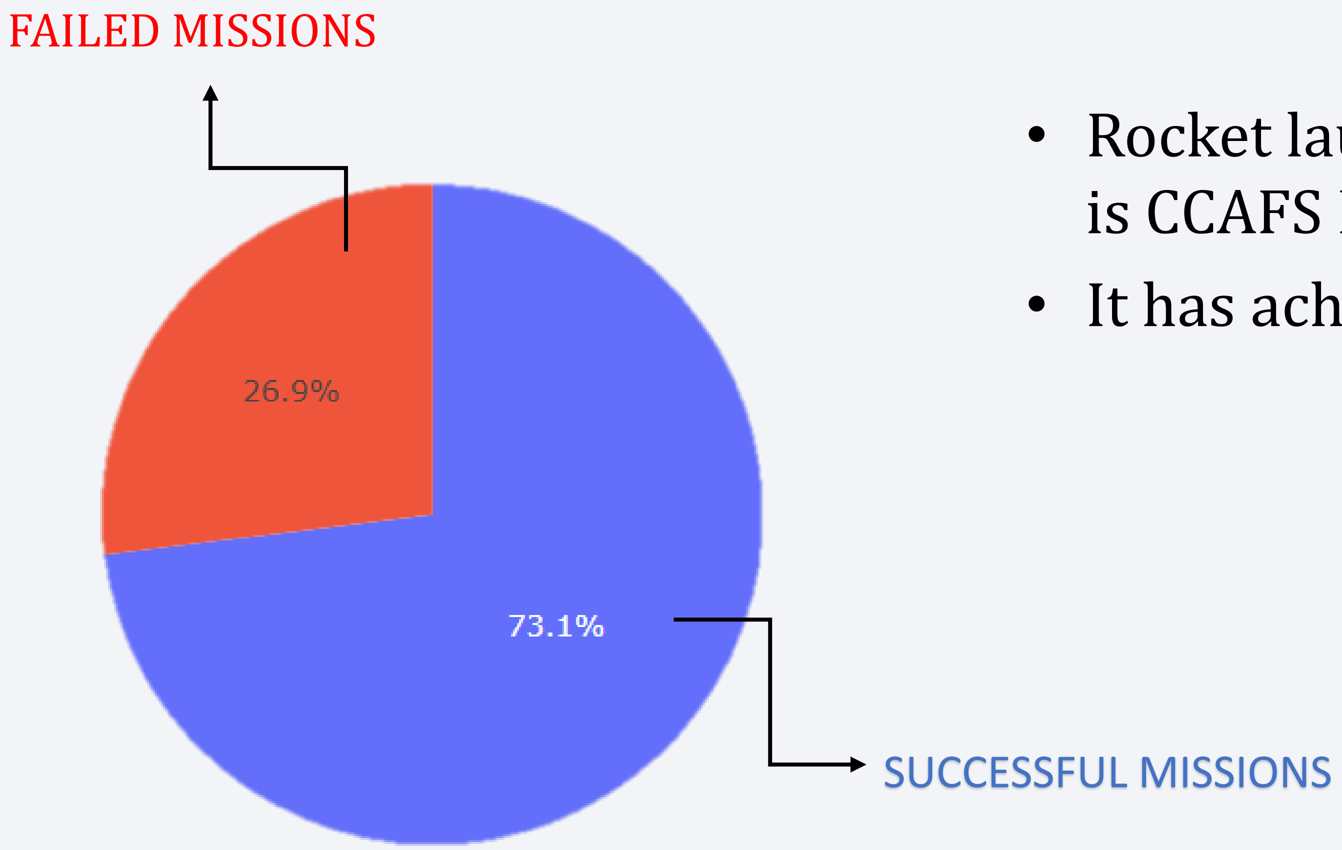
Build a Dashboard with Plotly Dash

LAUNCH SITE VS SUCCESS PIE CHART



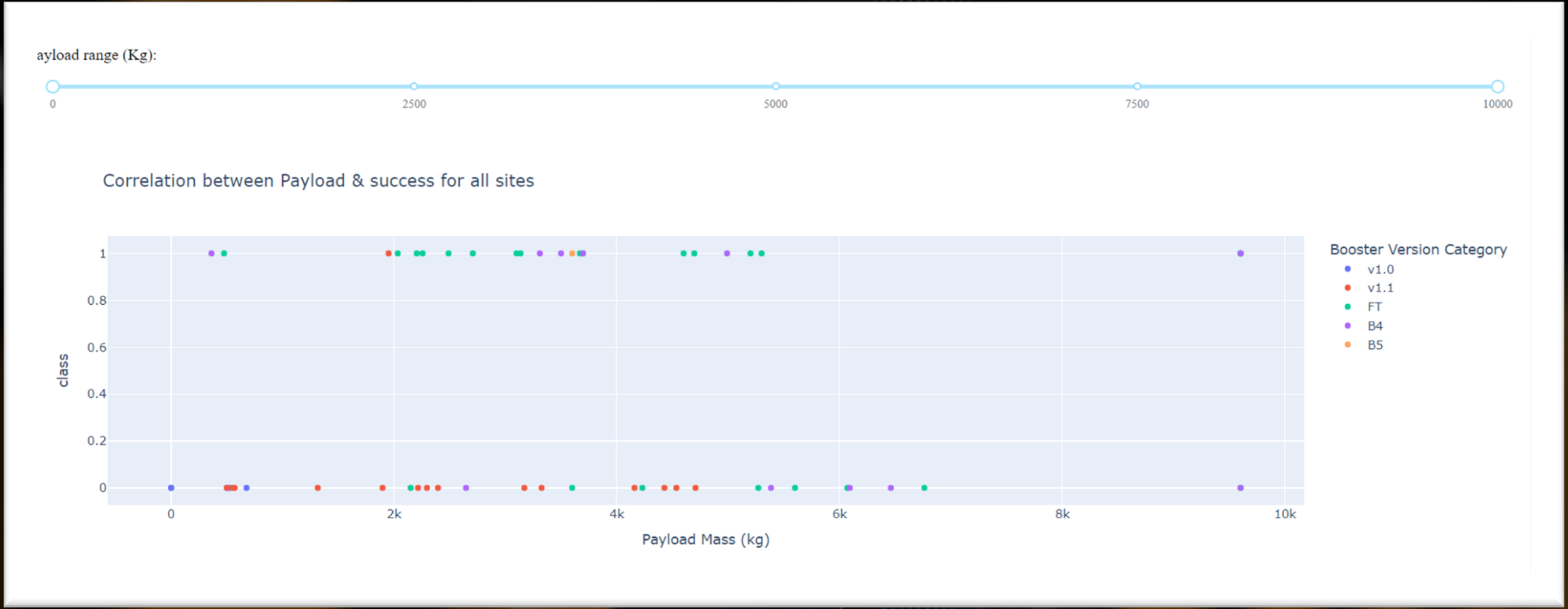
- Pie chart shows percentage success share of each Launch site.
- Most success was achieved at CCAFS LC 40 rocket launch site.

CCAFS LC 40 SUCCESS - FAILURE RATIO



- Rocket launch site with most success rate is CCAFS LC 40.
- It has achieved a success share of 73.1 %

RELATIONSHIP BETWEEN PAYLOAD AND SUCCESS FOR ALL SITES



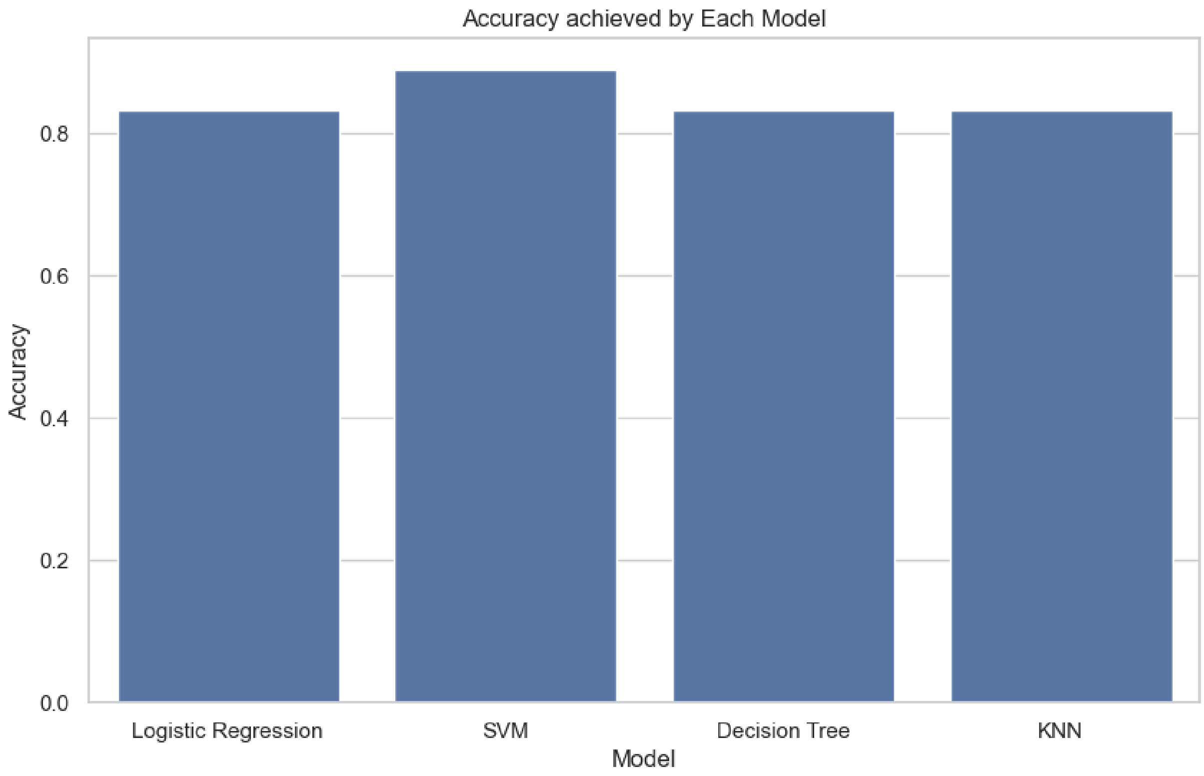
Scatter plot showing relationship between different booster version & success rate based on selected Payload Mass range.

- Success count when the payload is between 2000 & 5000 kg.
- Most boosters achieve success between this payload range.

Section 5

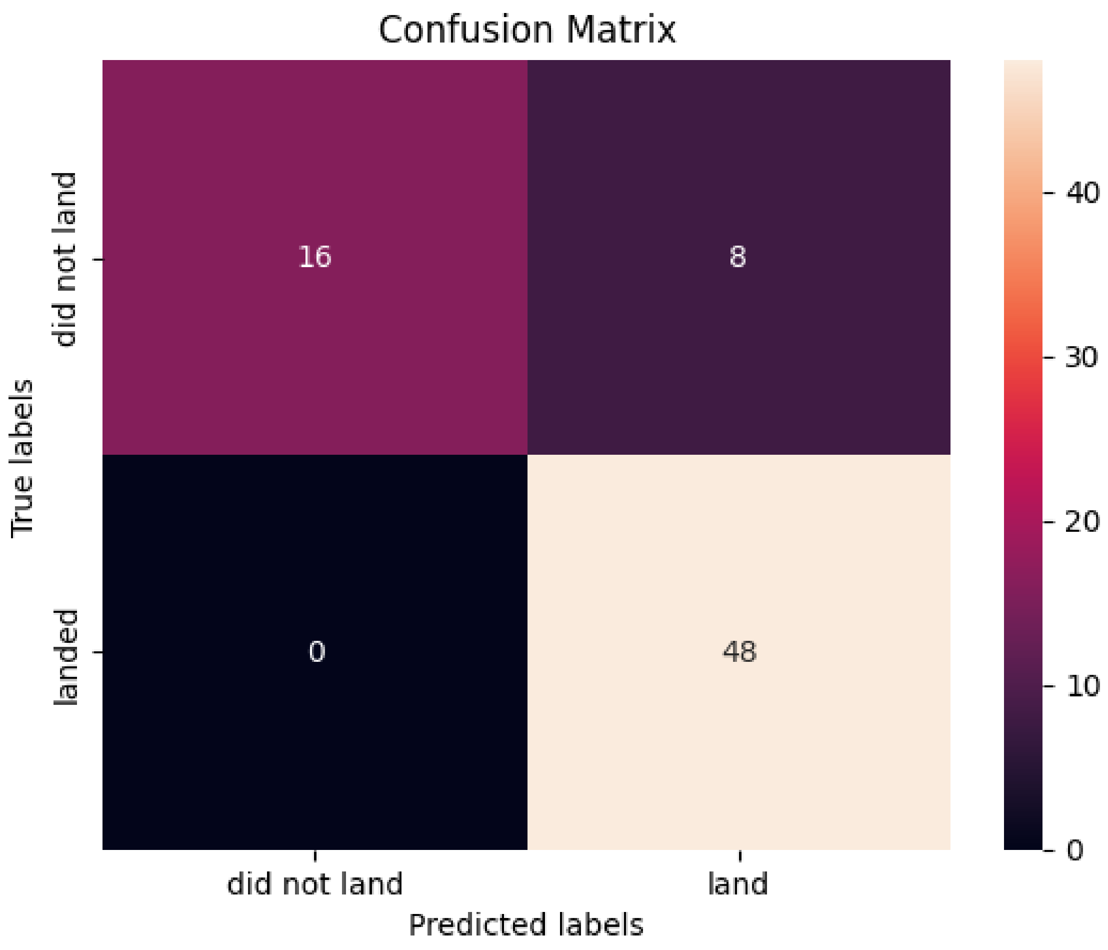
Predictive Analysis (Classification)

CLASSIFICATION ACCURACY



- SVM model made the most accurate prediction among all models.
- C, gamma, and kernel are hyperparameters that were tuned to optimize SVM model's performance.

CONFUSION MATRIX



The confusion matrix compares the actual target values with those predicted by the machine learning model.

Metrics	Value
Accuracy	88.89%
Precision	85.71%
Recall	100%
F1 Score	0.923

- **True Positives (TP)**: 16 instances where the model correctly predicted “did not land”.
- **False Negatives (FN)**: 8 instances where the model predicted “did not land” but the actual outcome was “landed”.
- **False Positives (FP)**: 0 instances where the model predicted “landed” but the actual outcome was “did not land”.
- **True Negatives (TN)**: 48 instances where the model correctly predicted “landed”

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{16 + 48}{16 + 48 + 0 + 8} = \frac{64}{72} \approx 0.89 \text{ or } 89\%$$

CONCLUSIONS

This analysis demonstrates that historical data can be effectively used to predict whether a Falcon 9 first stage will land successfully with high accuracy.

SVM Model predicted that first stage will successfully land and can be reused for future use.

As predicted probability of successful landing is above a certain threshold, we can estimate that the launch will be significantly cheaper due to reusability.

By integrating this prediction into cost models, SpaceX can better estimate launch costs and further optimize their operations.

CHALLENGES & RECOMMENDATION

DATA QUALITY: Ensuring high-quality and reliable sensor readings is crucial.

COMPLEXITY OF SYSTEMS: The complexity of rocket systems means that many variables need to be considered.

REAL-TIME PROCESSING: The system must process data quickly enough to provide timely predictions.

MODEL INTERPRETABILITY: Understanding why a model predicts failure can help in improving future launches.

CORRECT METHODS: Choose appropriate machine learning models or statistical techniques.

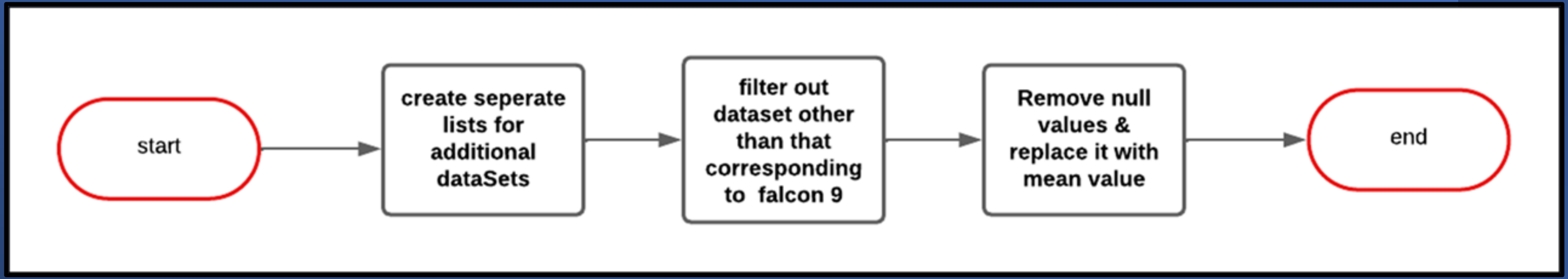
APPENDIX

VARIABLES	INFO
FlightNumber	Number to uniquely identify the flight.
PayloadMass	Mass of the Payload (shipment) in Kg.
Orbit	Each launch aims to a dedicated orbit. Orbit column contain different Type of orbits.
LaunchSite	Site where rockets are launched.
Flights	No of flights made by a rocket or booster.
GridFins	Innovative type of flight control surface used by SpaceX on their Falcon 9 rockets.
BoosterVersion	Booster version used by SpaceX rockets(e.g. Falcon 9).
Block	It refers to upgrades and iterations of SpaceX rockets.
Serial	serial number assigned to each rocket core or booster
Reused	Binary value indicating whether the rocket is been used or not.
Legs	Binary value indicating whether the rocket has used landing legs or not.
LandingPad	Serial number of landing pad used
ReusedCount	No of times booster has been used

DATA DICTIONARY

KEYWORDS	INFO
API	Application Programming Interface
EDA	Exploratory Data Analysis
LR	Logistic Regression
KNN	K-Nearest Neighbors (ML Algorithm)
SVM	Support Vector Machines (ML Algorithm)
DT	Decision Tree (ML Algorithm)
CSV	Coma Separated Values
BeautifulSoup	Python Library used for Web scrapping
DataFrame	data structure provided by the PANDAS library
features_one_hot	DataFrame with encoded variables.
float64	Special decimal data type

KEYWORDS	INFO
MarkerCluster	Folium object used for cluster marking
MousePosition	Folium object to get mouse’s position in the map.
PolyLine	Folium object to draw a line in map.
callback	functionality that are triggered by user interaction.
train_test_split	scikit learn library function used for splitting dataset.
GridSearchCV	tool used to tune hyperparameters of a ML model.
parameters	Settings tweaked to optimize model’s performance.
POI	Point of Interest
kernel	The type of kernel used in the SVM.
gamma	Value indicating the Influence each training set has on decision boundary.
C	The regularization parameter, which controls the trade-off between margin and misclassification error.




```sql
%%sql
SELECT
  CASE
    WHEN SUBSTR(Date, 6, 2) = '01' THEN 'January'
    WHEN SUBSTR(Date, 6, 2) = '02' THEN 'February'
    WHEN SUBSTR(Date, 6, 2) = '03' THEN 'March'
    WHEN SUBSTR(Date, 6, 2) = '04' THEN 'April'
    WHEN SUBSTR(Date, 6, 2) = '05' THEN 'May'
    WHEN SUBSTR(Date, 6, 2) = '06' THEN 'June'
    WHEN SUBSTR(Date, 6, 2) = '07' THEN 'July'
    WHEN SUBSTR(Date, 6, 2) = '08' THEN 'August'
    WHEN SUBSTR(Date, 6, 2) = '09' THEN 'September'
    WHEN SUBSTR(Date, 6, 2) = '10' THEN 'October'
    WHEN SUBSTR(Date, 6, 2) = '11' THEN 'November'
    WHEN SUBSTR(Date, 6, 2) = '12' THEN 'December'
  END AS Month_Name,
  Landing_Outcome,
  Booster_Version,
  Launch_Site
FROM SPACEXTABLE
WHERE SUBSTR(Date, 1, 4) = '2015'
AND Landing_Outcome LIKE '%Failure (drone ship)%';
```

```
FlightNumber        int64
Date               object
BoosterVersion     object
PayloadMass       float64
Orbit              object
LaunchSite         object
Outcome            object
Flights             int64
GridFins             bool
Reused               bool
Legs                 bool
LandingPad         object
Block             float64
ReusedCount         int64
Serial             object
Longitude         float64
Latitude          float64
dtype: object
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1

GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

Thank you!