IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Muhammad Faiz Amir
Aththufail
December 12, 2022

Outline

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

Executive Summary

- The data was collected through various sources, such as SpaceX REST API and Wiki pages.
- The data includes data for the Falcon 1 booster which is unnecessary for predicting the Falcon 9 launches outcome, therefore it was removed.
- The PayloadMass data had some missing values, we then dealing these missing values by replacing it with the mean of PayloadMass data.
- Then we visualized the data by building an interactive dashboard.
- We then train machine learning models, such as Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbors and find the best hyperparameters to predict if SpaceX will reuse the first stage.
- The result shows that all models perform similarly considering the similar results on the accuracy for the test data

Introduction

- SpaceX is one of the most successful companies that are making space travel affordable for everyone. SpaceX's accomplishments include, sending spacecraft to the International Space Station, providing satellite internet access using a satellite internet constellation called Starlink, and sending manned missions to space. One reason SapceX can do this is because SpaceX's rockets can reuse the first stage, therefore the rocket launches are relatively inexpensive which only cost 62 million dollars compared to other companies which cost upwards of 165 million dollars.

- By determining whether the first stage will land or not, we can determine the cost of a launch

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from various sources, such as SpaceX REST API and Wiki pages
- Perform data wrangling
  - Landing outcomes data was processed by separating bad outcomes and successful outcomes then converting it to numerical values (0 and 1)
- Perform exploratory data analysis (EDA) using visualization and SQL

Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Normalized the features data
  - Splitted data into training set and test set
  - Trained machine learning models using the training set
  - Tuned the hyperparameters and looked for the best hyperparameters
  - Performed the models using the test set
  - Calculated the accuracy score

Data Collection

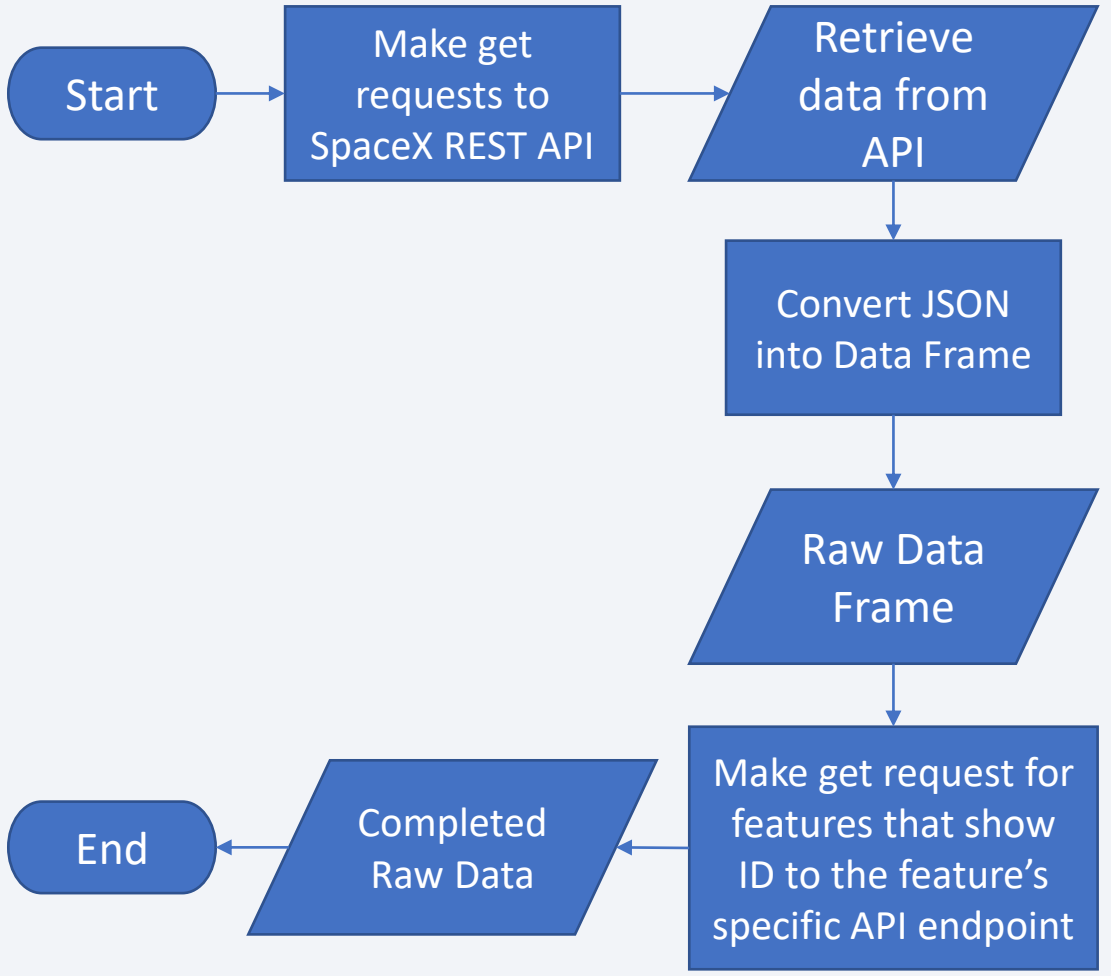
- The data was collected from the SpaceX REST API and Wiki Pages.
- The data from SpaceX REST API was collected by performing HTTP GET Request to the SpaceX REST API which returns JSON format response then converting it to Data Frame
- The data from Wiki Pages was collected by performing GET Request to the URL and then webscrapping it using BeautifulSoup

https://github.com/faizamir123/IIBM_Capstone

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

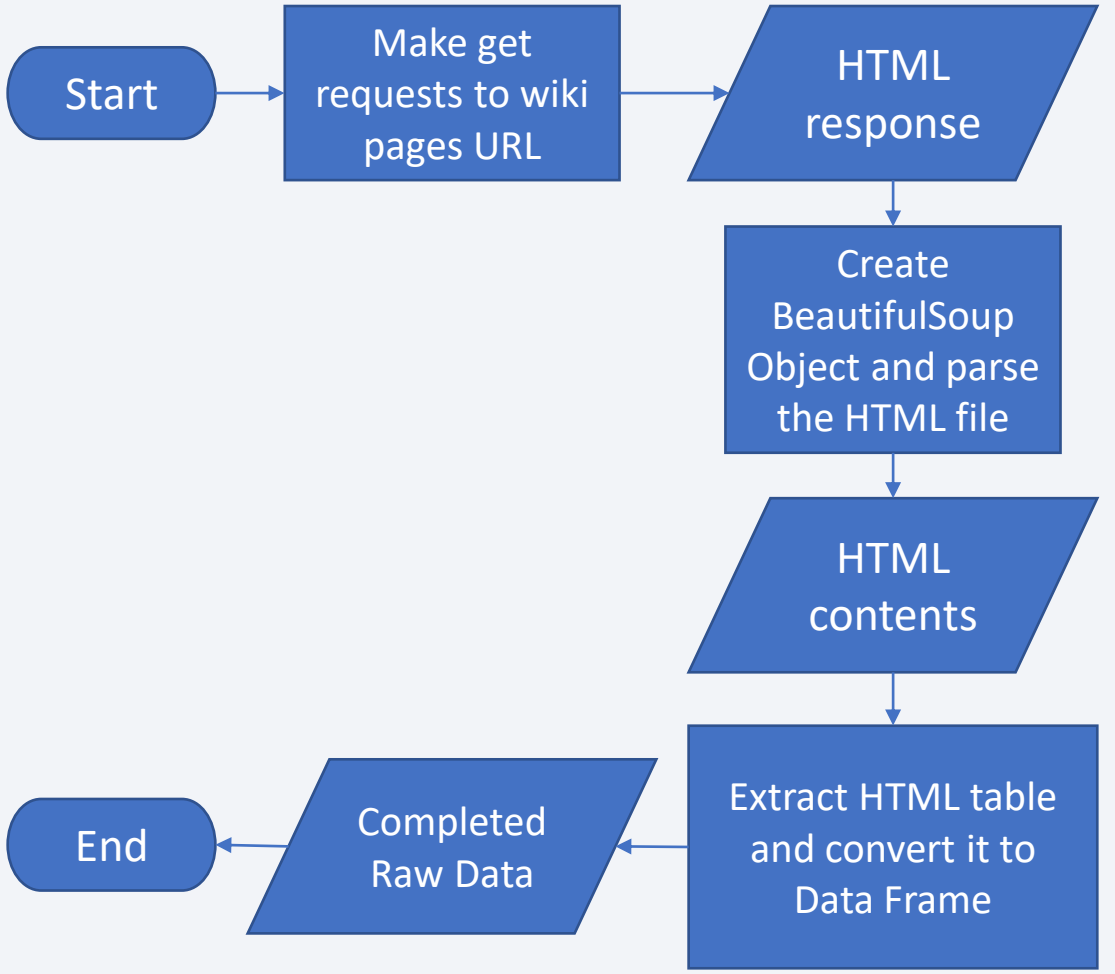
- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

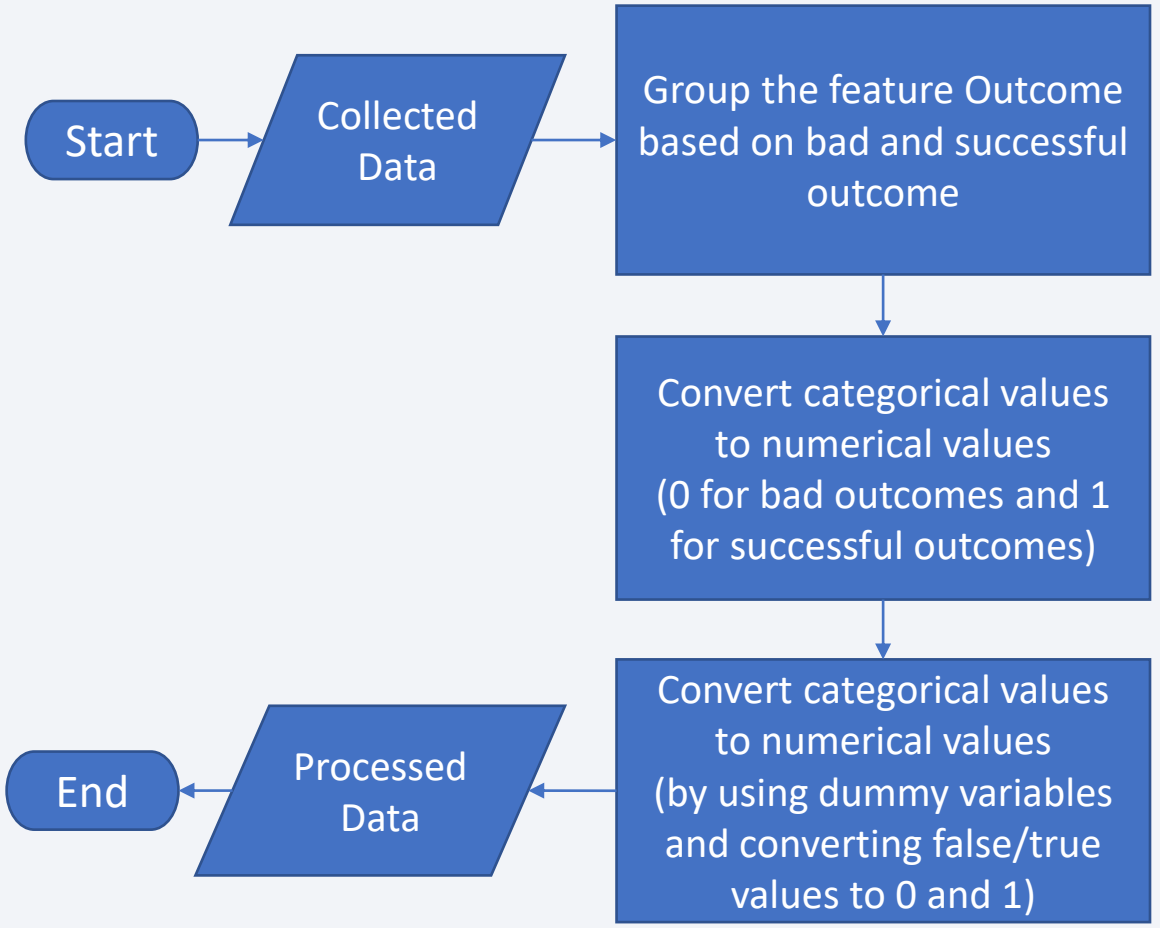
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



Data Wrangling

- The collected data has some features that need to be processed, for example the column feature **Outcome** which has to be grouped between the bad outcome and successful outcome.
- Bad outcomes include False ASDS, False Ocean, False RTLS, None ASDS, None None
- Successful outcomes include True ASDS, True RTLS, and True Ocean
- Make dummy variables to convert categorical features like orbits, launch site, landing pad, and serial

https://github.com/faizamir123/IIBM_Capstone



EDA with Data Visualization

- Scatter Plot: to see the relationship between two features
- Bar Chart: to visualize the frequency (in this case the success rate) of a feature
- Line Chart: to visualize the trend over the time

https://github.com/faizamir123/IIBM_Capstone

EDA with SQL

- Using SELECT DISTINCT statement to display the name of unique launch sites

```
%%sql
SELECT DISTINCT(Launch_site) FROM SPACEXTBL
```

- Using WHERE clause and LIKE clause to filter launch sites begin with CCA

```
%%sql
SELECT * FROM SPACEXTBL WHERE Launch_site LIKE 'CCA%'
LIMIT 5
```

- Using SUM function to display the total Payload Mass and filter the customer to NASA (CRS)

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

- Using AVG function to display the average Payload Mass and filter the booster version that begin with F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE Booster_Version LIKE 'F9 v1.1%'
```

- Using where to choose only the successful landing outcome, substr function to extract the date and MIN function to find the first successful landing outcome

```
%%sql
SELECT MIN(substr(Date,7,4) || '-' || substr(Date,4,2) || '-' || substr(Date,1,2)) as date_yyyymmdd FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (ground pad)'
```

EDA with SQL

- Display the name of the booster version and using WHERE clause to filter 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 "Landing _Outcome" = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_<6000)
```

- Using Common Table Expressions to display the total number of successful and failure missions outcomes

```
%%sql
WITH CTE1 AS (SELECT COUNT(Mission_Outcome) AS SUCCESS FROM SPACEXTBL
WHERE Mission_Outcome LIKE 'Success%'), CTE2 AS (SELECT COUNT(Mission_Outcome) AS FAILURE FROM SPACEXTBL
WHERE Mission_Outcome LIKE 'Failure%')

SELECT SUCCESS, FAILURE FROM CTE1,CTE2
```

- Using a subquery and MAX function to list the names of the booster versions which have carried the maximum Payload Mass

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

EDA with SQL

- Using substr function to extract the months and year and using CASE to convert the month number to month name then display the month names, failure landing outcomes in drone ship, booster version, and launch site for months in year 2015

```
%%sql
SELECT CASE substr(Date, 4, 2)
WHEN '01' THEN 'January'
WHEN '02' THEN 'February'
WHEN '03' THEN 'March'
WHEN '04' THEN 'April'
WHEN '05' THEN 'May'
WHEN '06' THEN 'June'
WHEN '07' THEN 'July'
WHEN '08' THEN 'August'
WHEN '09' THEN 'September'
WHEN '10' THEN 'October'
WHEN '11' THEN 'November'
WHEN '12' THEN 'December'
END AS MONTH, "Landing _Outcome", Booster_Version, Launch_site
FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE "Failure%" AND SUBSTR(Date, 7, 4)='2015'
```

- Using COUNT function and WHERE clause to count the successful landing outcomes between the date 04-06-2010 and 20-03-2017, using GROUP BY to group the data based on landing outcomes, and using ORDER BY to rank it in descending order

```
%%sql
SELECT "Landing _Outcome", COUNT("Landing _Outcome") FROM SPACEXTBL
WHERE Date BETWEEN '04-06-2010' AND '20-03-2017' AND "Landing _Outcome" LIKE 'Success%'
GROUP BY "Landing _Outcome"
ORDER BY COUNT("Landing _Outcome") DESC
```

Build an Interactive Map with Folium

- Marker objects are added to folium map to mark all the launch sites' locations
- Circle objects are added to folium map to mark the area of the launch sites' locations
- Marker Cluster objects are used to cluster/group the number of events that occurred in a certain area in this case success/failed launches for each site
- Line objects are used to show distances between a launch site to its proximities

https://github.com/faizamir123/IIBM_Capstone

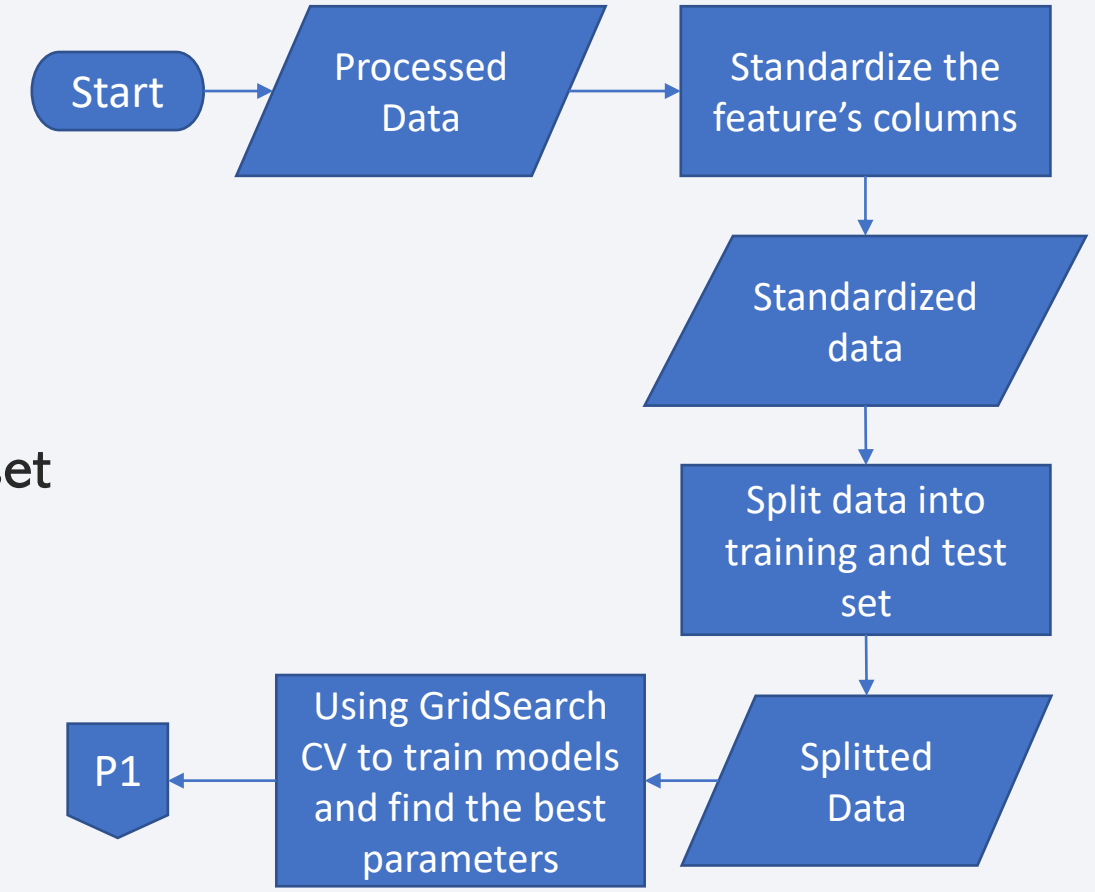
Build a Dashboard with Plotly Dash

- There are two types of plots that have been added to the dashboard, pie chart and scatter plot
- Pie charts are used to show the proportion of the number of successful launches between launch sites and the proportion of successful and failed launches on each site
- Scatter plots are used to show the relationship between Payload Mass and the outcome of the launches

https://github.com/faizamir123/IIBM_Capstone

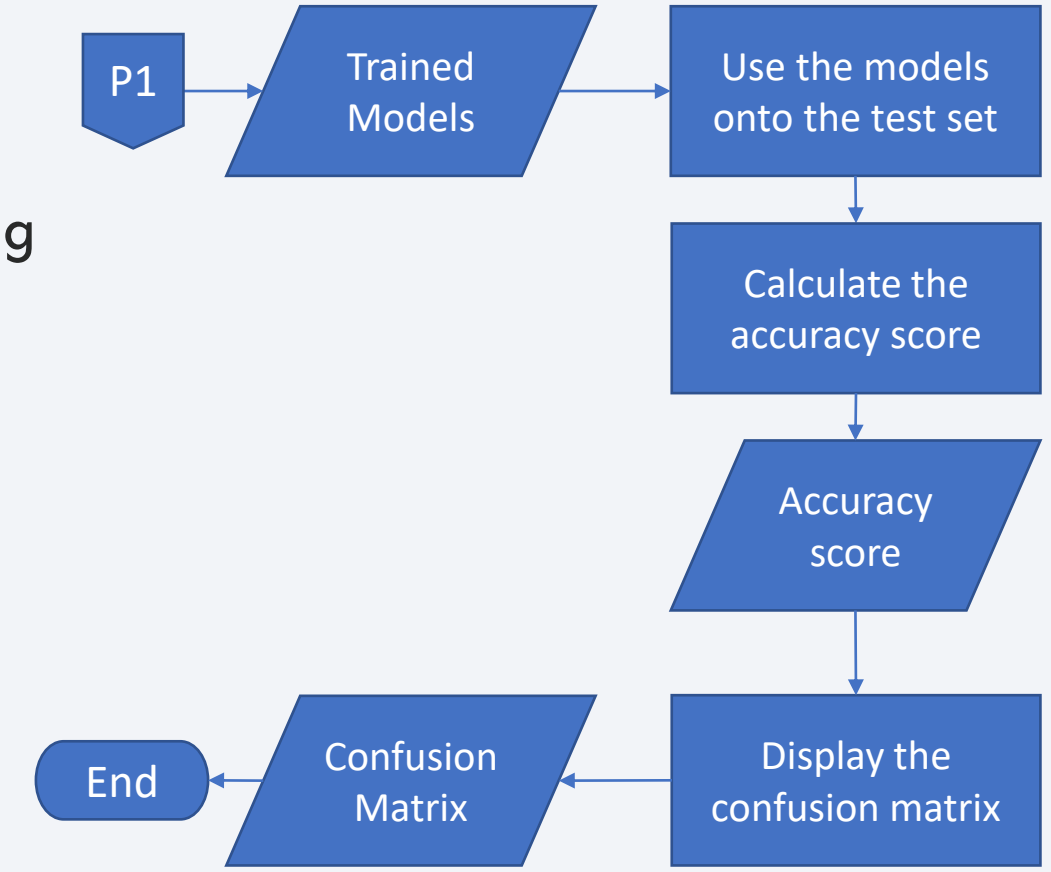
Predictive Analysis (Classification)

- Standardize the feature's columns using standardscaler
- Split the data into training and test set



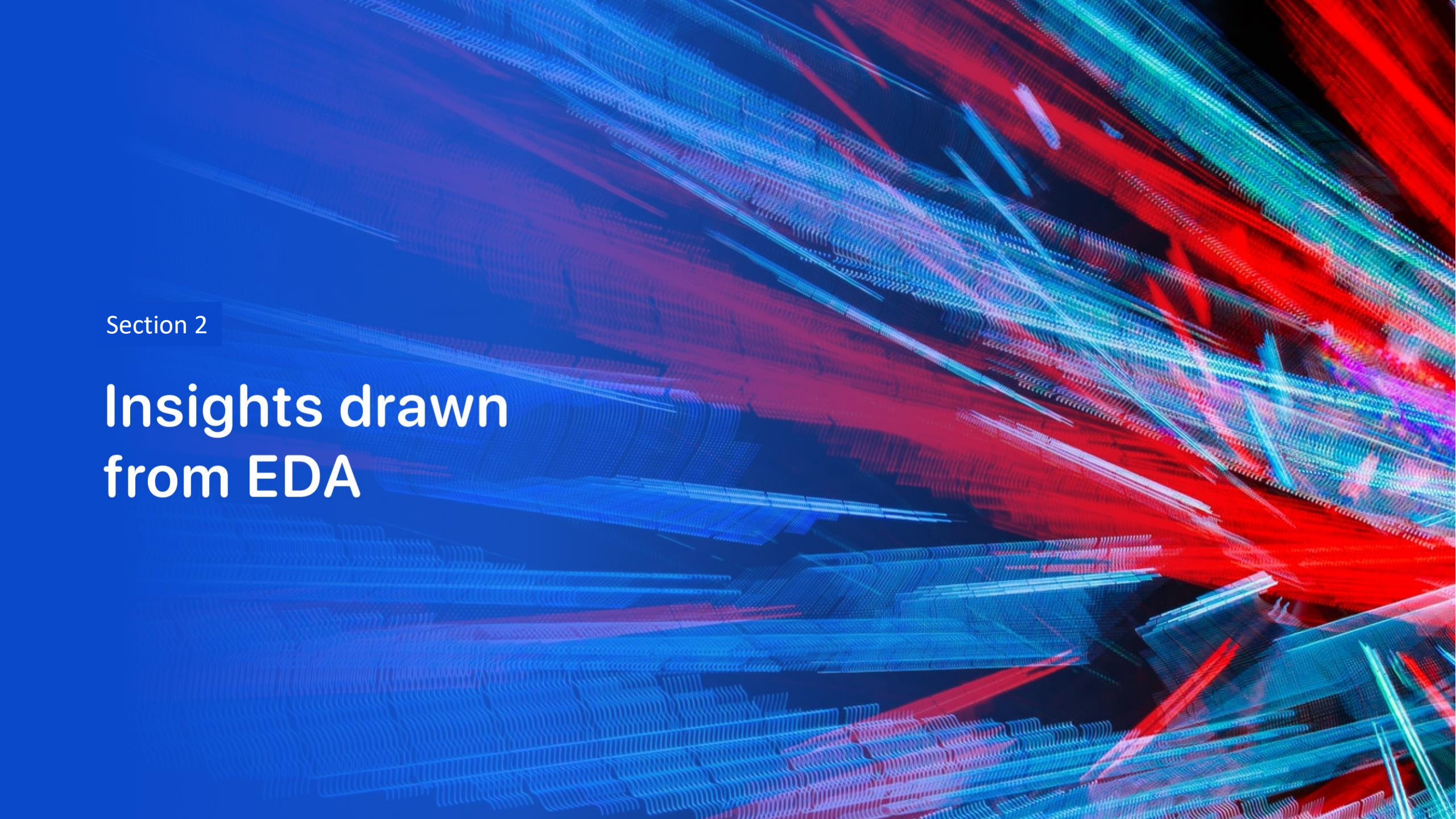
Predictive Analysis (Classification)

- Using GridSearchCV to automate the process of training models and finding the best parameters and hyperparameters for each model
- Calculate the accuracy score on the test set
- Display the confusion matrix



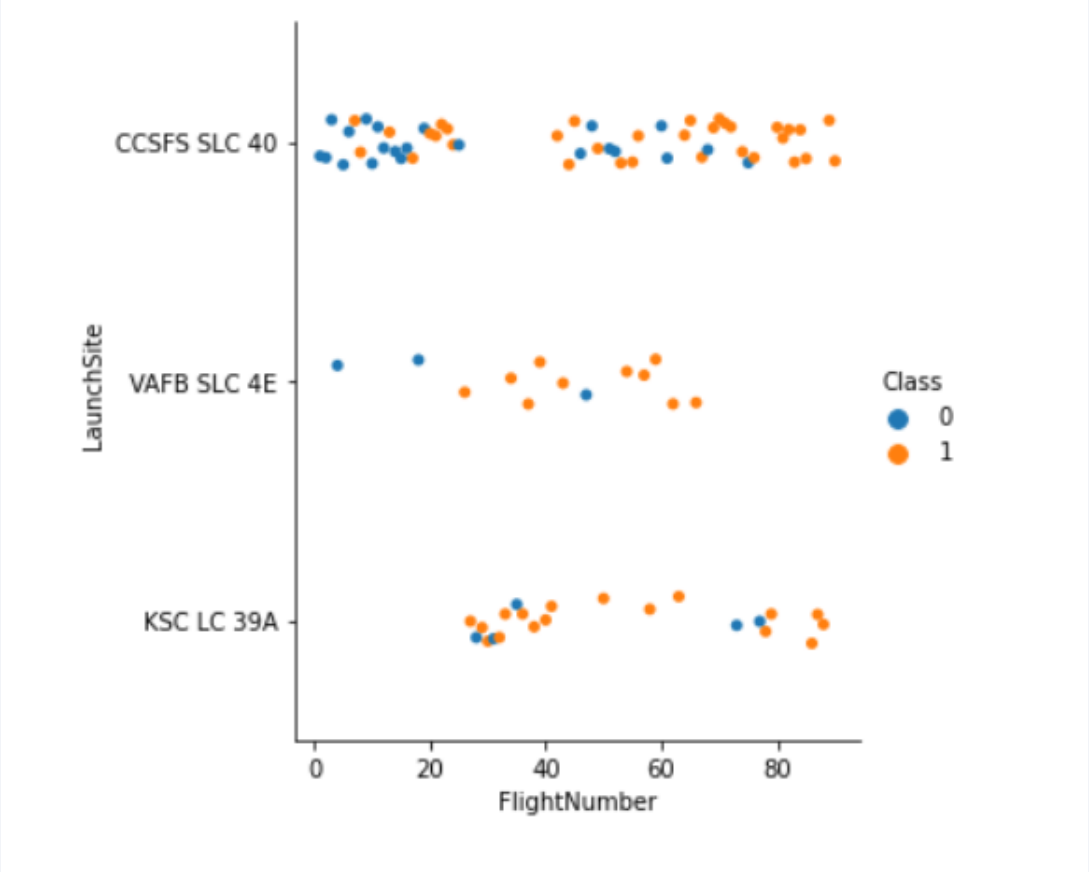
Results

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


Section 2
Insights drawn from EDA

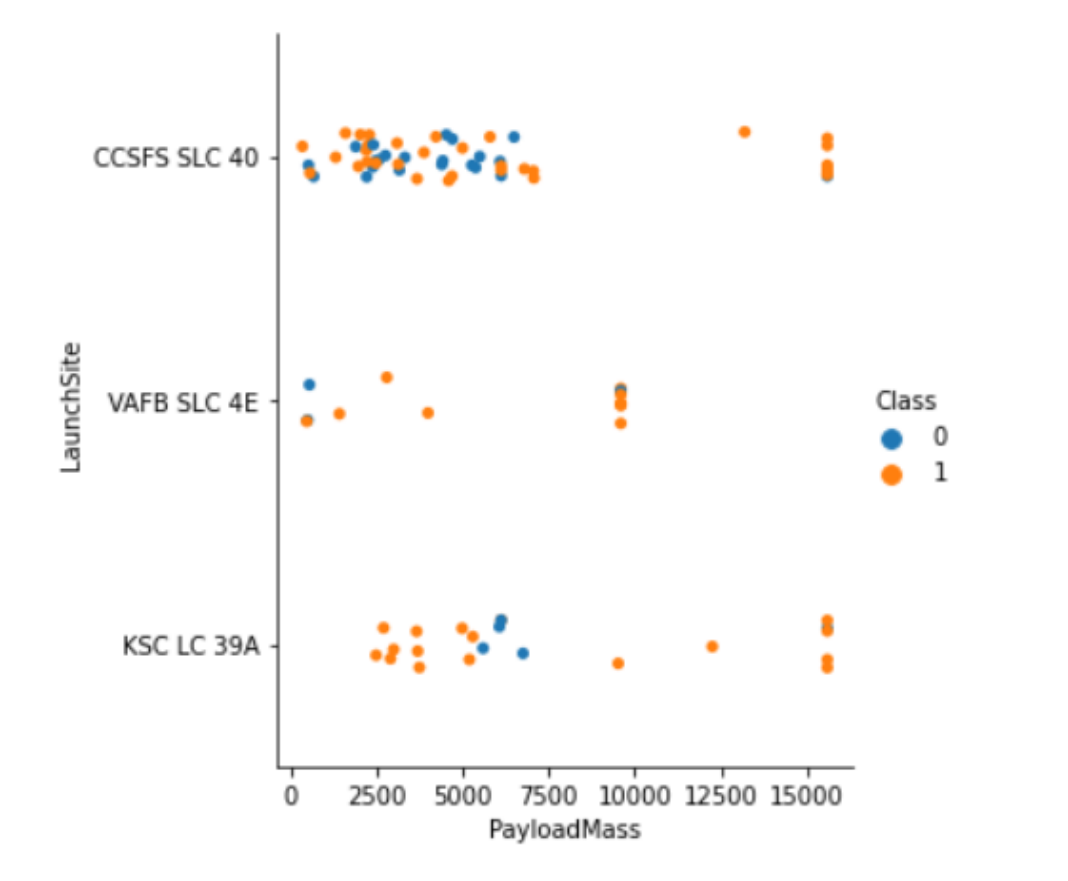
Flight Number vs. Launch Site

- On each launch sites as the flight number increases, the first stage is more likely to land successfully.
- Many launches took place in CCSFS SLC 40 launch site



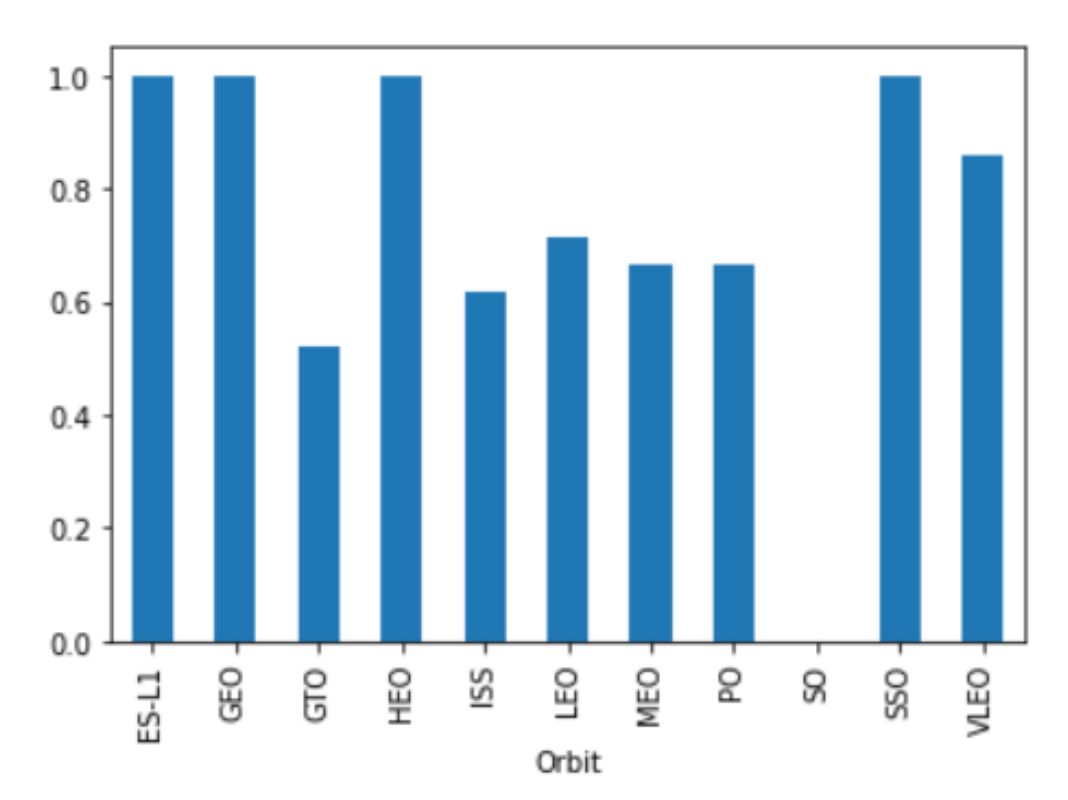
Payload vs. Launch Site

- There are no rockets launched for heavypayload mass (greater than 10000) for the VAFB-SLC 4E launch site.
- The rockets with payload greater than 7500 are more likely to land the first stage successfully



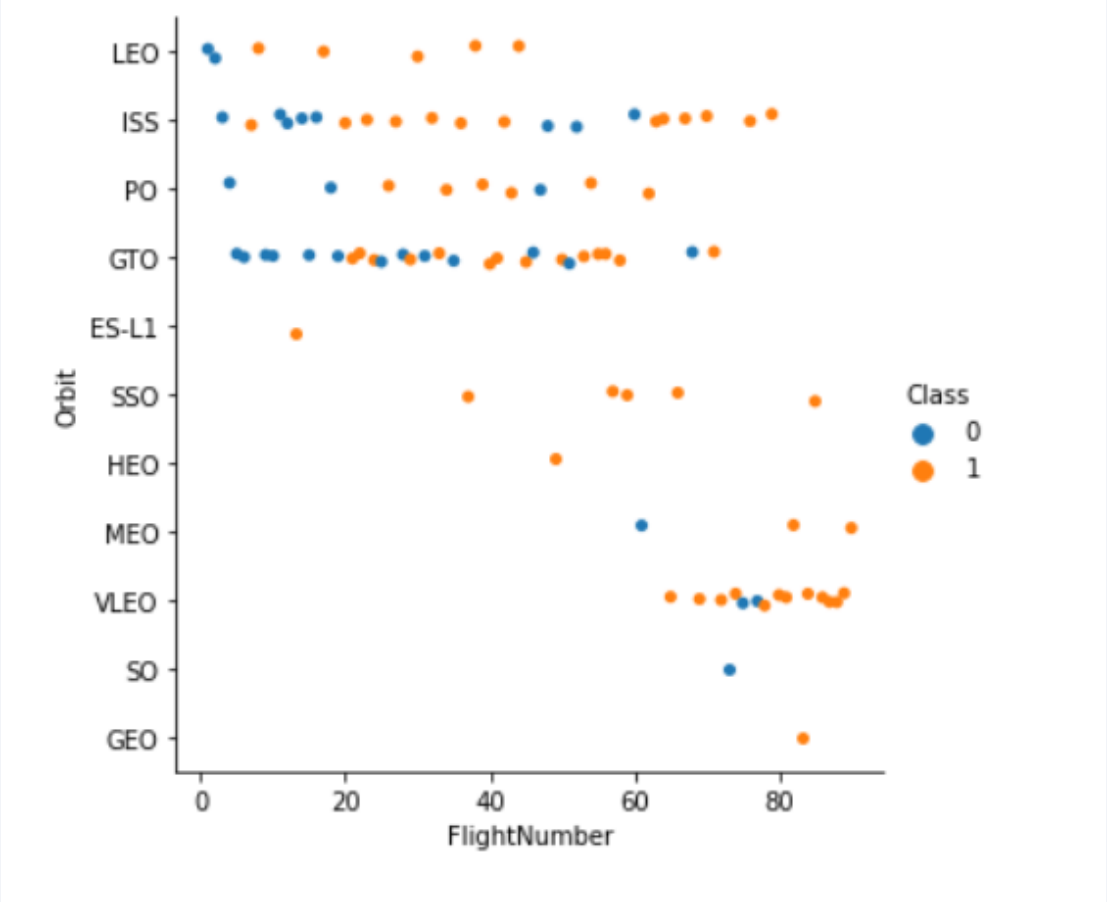
Success Rate vs. Orbit Type

- The orbit ES-L1, GEO, HEO, and SSO have the highest successful rate, which is 1
- Whereas the orbit with the lowest successful rate is SO, with the successful rate of 0



Flight Number vs. Orbit Type

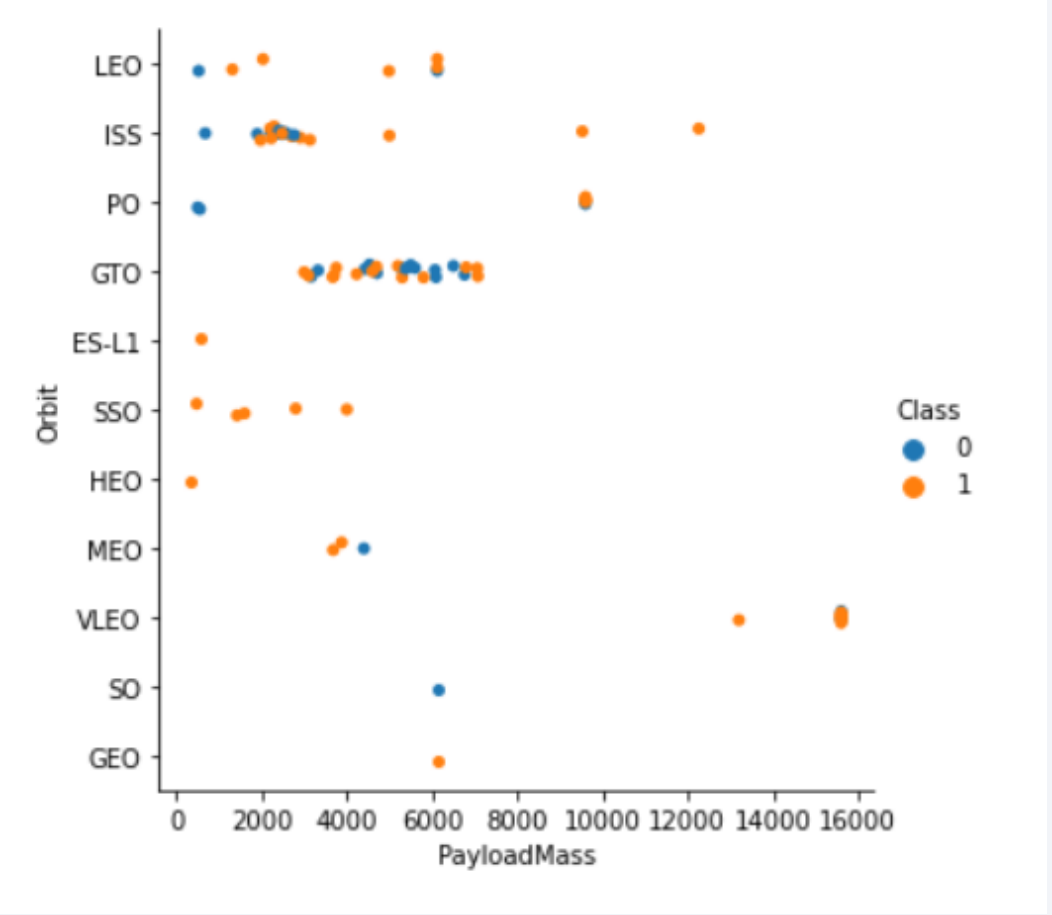
- There seems to be no relationship between flight number in GTO orbit
- In LEO and PO orbit the success appears to be related to the flight number, as the flight number increases, the first stage is more likely to land successfully



Payload vs. Orbit Type

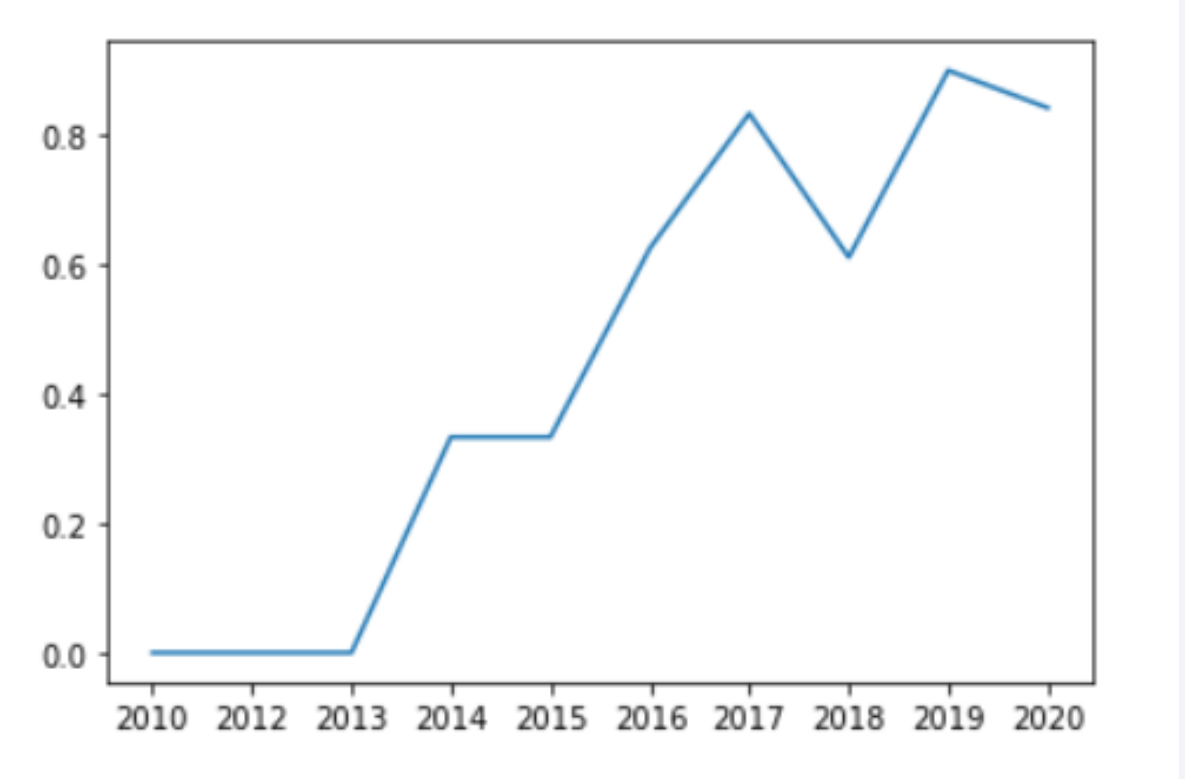
- With heavier payload mass, the first stage is more likely to land successfully for LEO, ISS, and PO orbit.

- However, we couldn't distinguish the relationship for GTO orbit



Launch Success Yearly Trend

- The success rate kept increasing from 2013 through 2020, although there was a slight decrease in 2018, but overall the success rate kept increasing



All Launch Site Names

- Using DISTINCT function, we return the names of the unique launch sites

```
%%sql
SELECT DISTINCT(Launch_site) FROM SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

- Using WHERE clause and LIKE clause, we filter the launch sites with the name begin with CCA and display only the first 5 records using LIMIT clause

```
%%sql
SELECT * FROM SPACEXTBL WHERE Launch_site LIKE 'CCA%'
LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Using the WHERE clause we filter the customer to NASA (CRS) and we calculate the total payload mass using the SUM function

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

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Using the WHERE and LIKE clause we filter the booster version that start with F9 v1.1
- Using the AVG function we calculate the average payload mass carried by booster version F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- We extract the date using substr function
- Using WHERE clause we filter the landing outcome so we have the success landing in ground pad
- Using MIN function we search for the first record of the successful landing outcome in ground pad

```
%%sql
SELECT MIN(substr(Date,7,4) || '-' || substr(Date,4,2) || '-' || substr(Date,1,2)) as date_yyyymmdd FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (ground pad)'
```

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

date_yyyymmdd
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Using WHERE clause we filter the landing outcome to Success (drone ship) and payloadmass greater than 4000 and less than 6000
- We obtain the list of booster versions that land successfully in drone ship and have payload mass greater than 4000 but less than 6000

```
%%sql
SELECT Booster_version FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_<6000)
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Using Common Table Expressions (CTE), we return the number of successful and failure mission outcome

```
%%sql
WITH CTE1 AS (SELECT COUNT(Mission_Outcome) AS SUCCESS FROM SPACEXTBL
WHERE Mission_Outcome LIKE 'Success%'), CTE2 AS (SELECT COUNT(Mission_Outcome) AS FAILURE FROM SPACEXTBL
WHERE Mission_Outcome LIKE 'Failure%')

SELECT SUCCESS, FAILURE FROM CTE1,CTE2
```

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

SUCCESS	FAILURE
100	1

Boosters Carried Maximum Payload

- Using a subquery we are able to use MAX function inside WHERE clause to search for the names of the booster versions which have carried the maximum payload mass

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

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

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- Using CASE we assign the month name to the month number
- Using WHERE clause we filter the landing outcome so it returns the failure landing in drone ship at 2015
- We display the record list containing the month name, landing outcome, booster version, and launch site

```
%%sql
SELECT CASE substr(Date, 4, 2)
WHEN '01' THEN 'January'
WHEN '02' THEN 'February'
WHEN '03' THEN 'March'
WHEN '04' THEN 'April'
WHEN '05' THEN 'May'
WHEN '06' THEN 'June'
WHEN '07' THEN 'July'
WHEN '08' THEN 'August'
WHEN '09' THEN 'September'
WHEN '10' THEN 'October'
WHEN '11' THEN 'November'
WHEN '12' THEN 'December'
END AS MONTH, "Landing _Outcome", Booster_Version, Launch_site
FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE "Failure%" AND SUBSTR(Date, 7, 4)='2015'
```

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

MONTH	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

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

- Using the COUNT function we return the number of landing outcome
- We filter the data using WHERE and BETWEEN clause so it return the result between the given date range
- Using GROUP BY, we group the result based on the landing outcome
- We order the count data descending uasing ORDER BY and DESC, so it return the rank of landing outcomes

```
%%sql
SELECT "Landing _Outcome", COUNT("Landing _Outcome") FROM SPACEXTBL
WHERE Date BETWEEN '04-06-2010' AND '20-03-2017' AND "Landing _Outcome" LIKE 'Success%'
GROUP BY "Landing _Outcome"
ORDER BY COUNT("Landing _Outcome") DESC
```

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

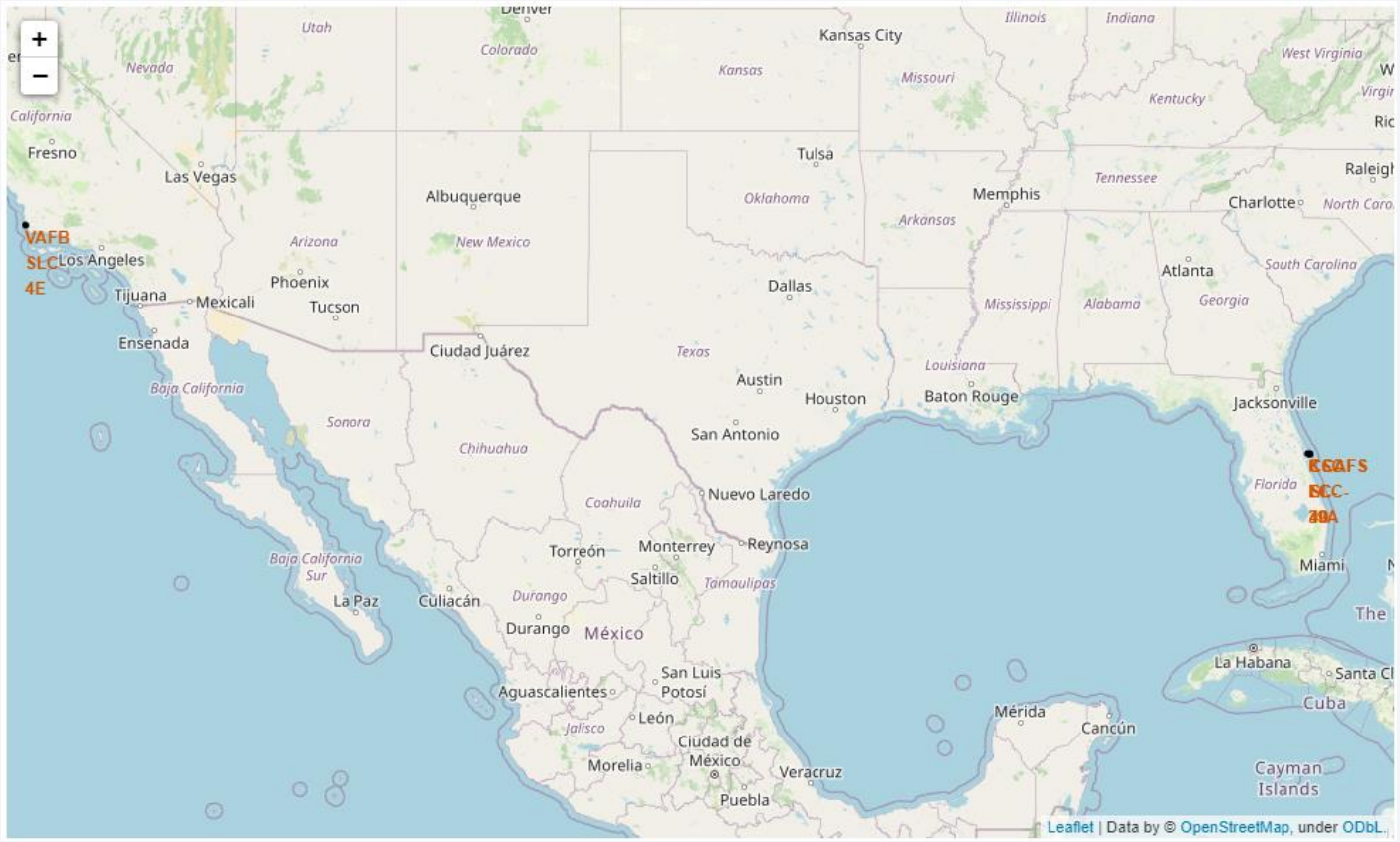
Landing _Outcome	COUNT("Landing _Outcome")
Success	20
Success (drone ship)	8
Success (ground pad)	6

Section 3

Launch Sites Proximities Analysis

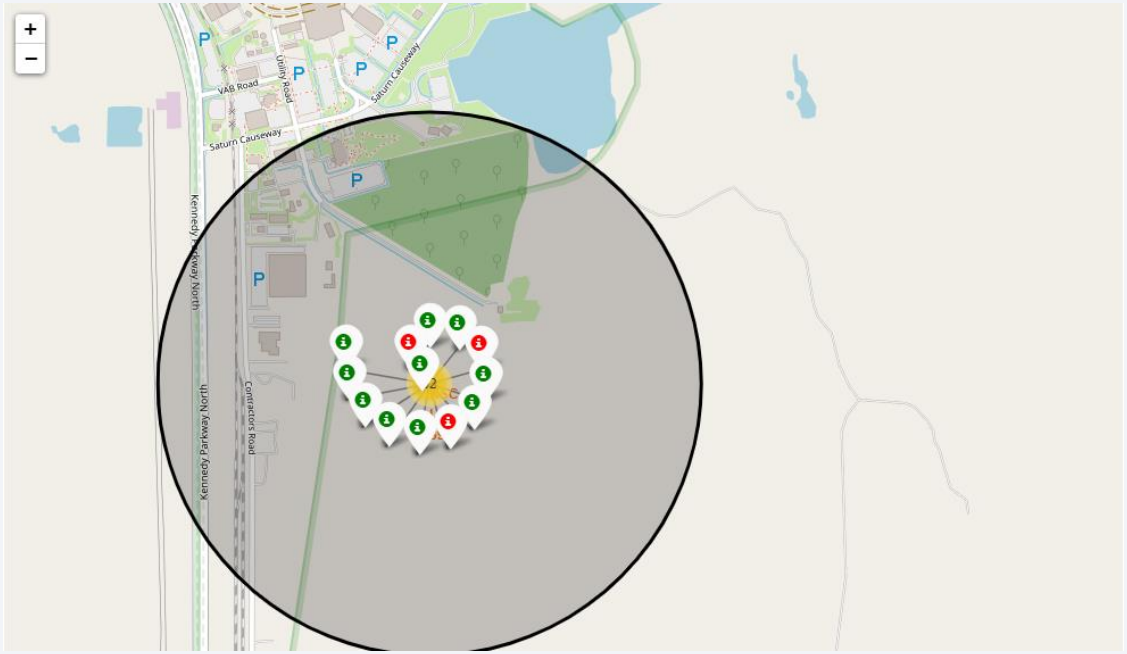
Launch Sites Location

- As we can see in the following picture the launch sites are located near the coastline
- This could be because it is safer to launch the rockets to the sea



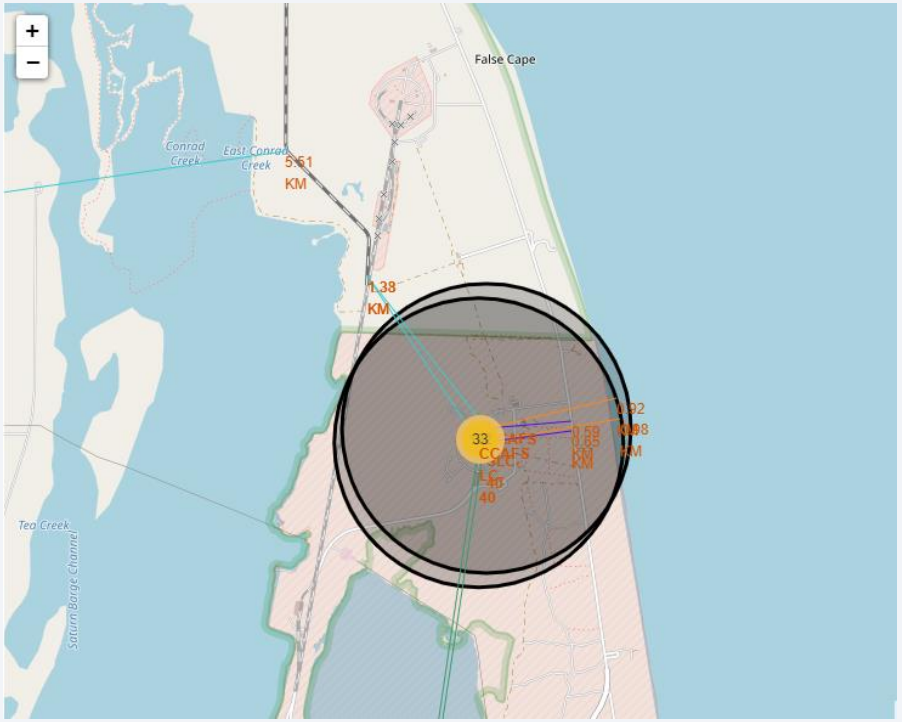
Launch Outcomes

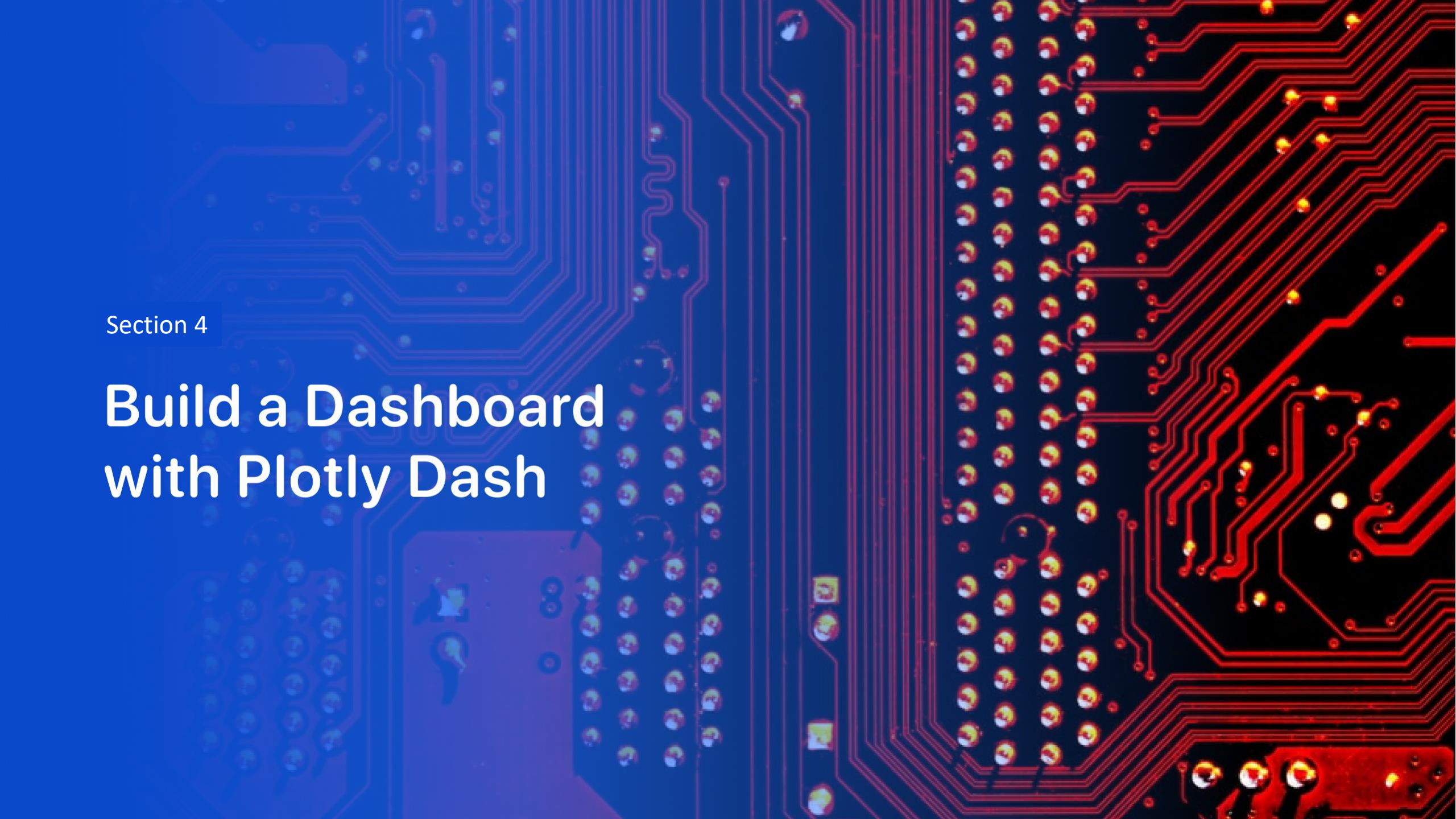
- We could see that the color red represent the failed launch outcomes, whereas the color green represent the successful launch outcomes
- The following picture is an example of launch outcomes in KSC LC-39A launch site
- The launch site KSC LC-39A have a pretty high successful launch outcome rate



Distance Between Launch Sites to its Proximities

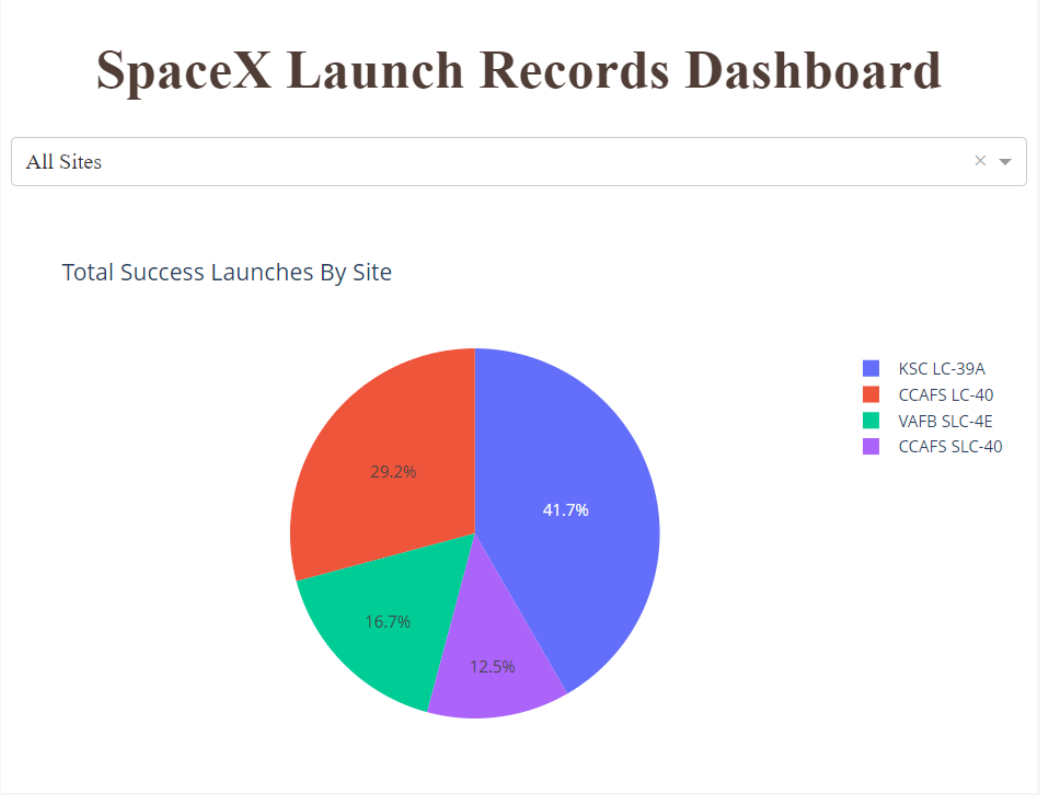
- The following picture shows the distance between launch site and its proximities such as the closest coastline, city, railway and highway.
- The distance between launch site and the closest coastline is represented by the orange line
- The distance between launch site and the closest railway is represented by the cyan line
- The distance between launch site and the closest highway is represented by the purple line
- The distance between launch site and the closest city is represented bby the green line




Section 4
Build a Dashboard with Plotly Dash

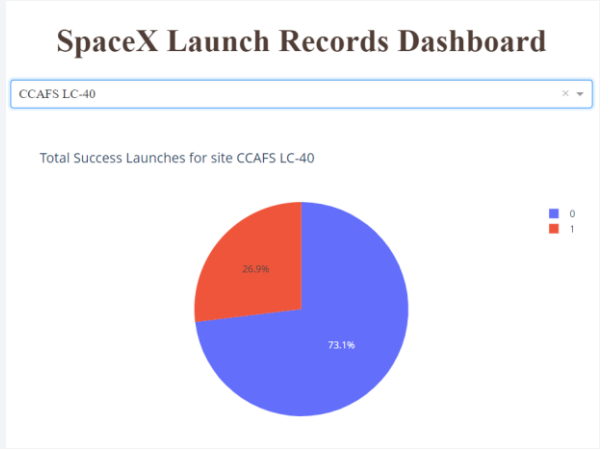
Total Success Launches By Site

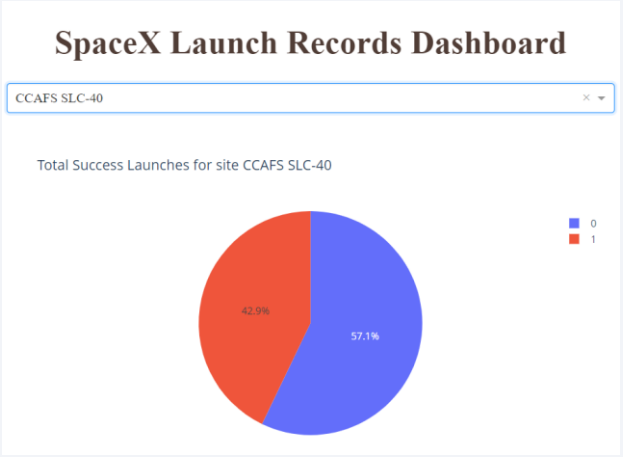
- The given pie chart tells us that the launch site KSC LC-39A have the highest number of successful launches followed by CCAFS LC-40, VAFB SLC-4E, and the last is CCAFS SLC-40

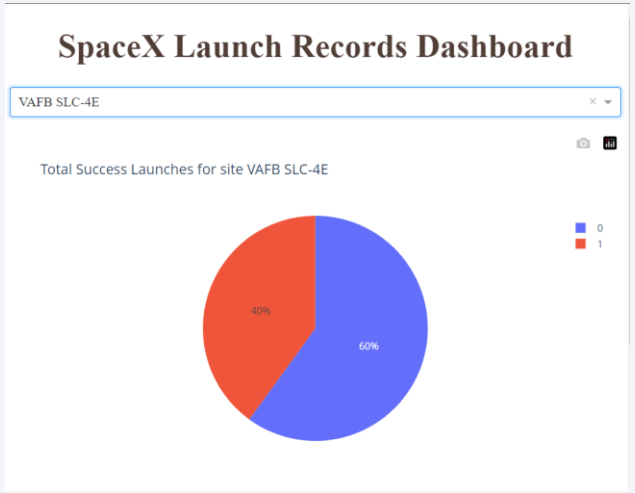


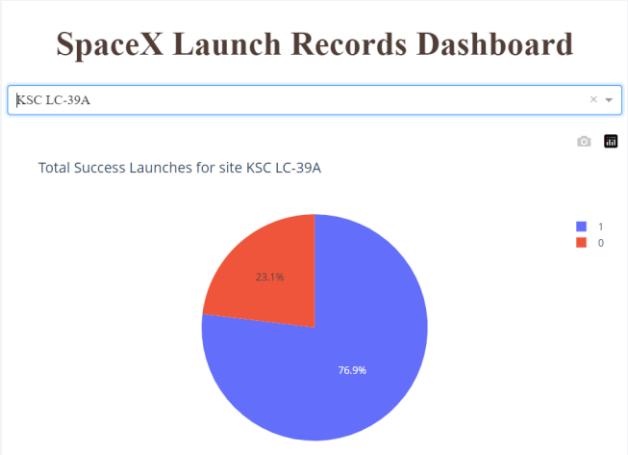
<Dashboard Screenshot 2>

- From the following pie charts we can see that launch site KSC LC-39A have the highest successful launch outcomes ratio with the ratio of 76.9%



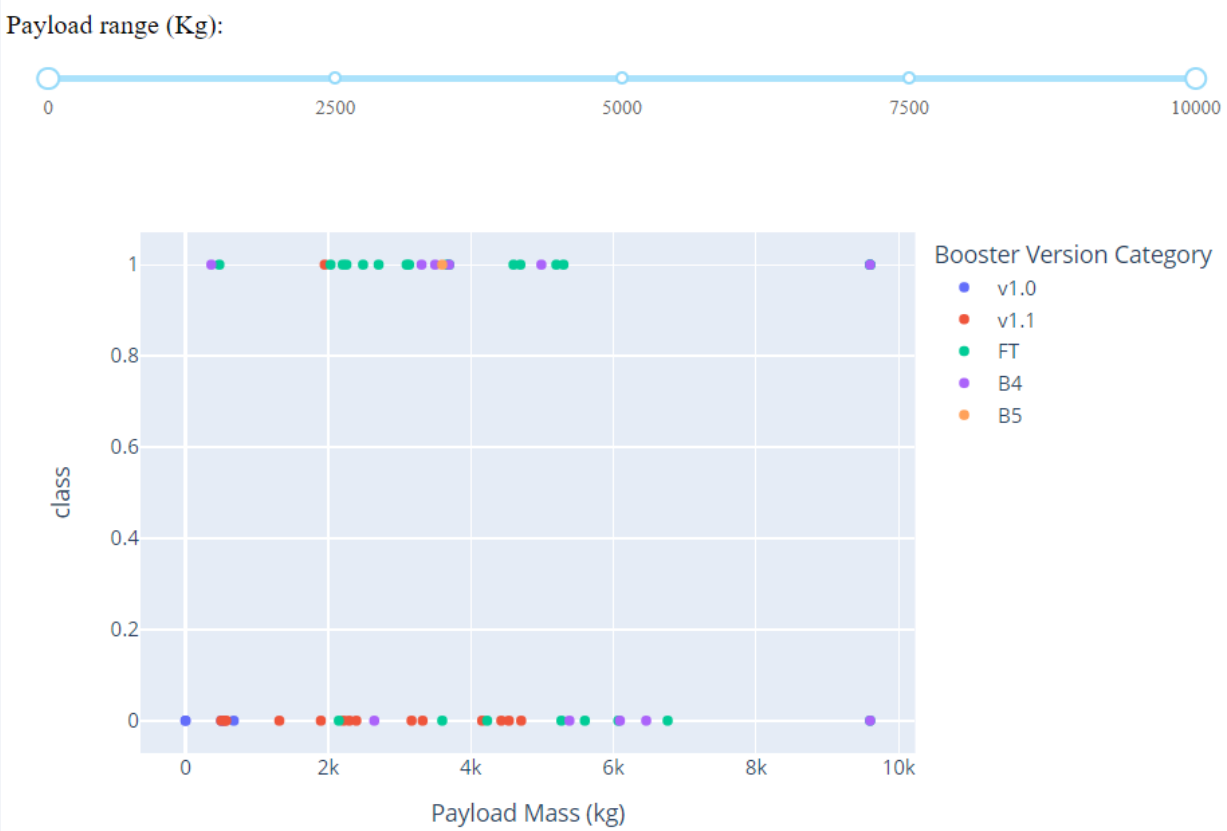






Payload vs. Launch Outcome scatter plot for All Sites

- Based on the following scatter plot about payload vs. Launch Outcome for all sites, we can see that the booster version FT with the payload mass between 2000 until 6000 kg have the highest successful rate
- Whereas the booster version v1.1 with the payload mass between 0 until 6000 have the lowest successful rate

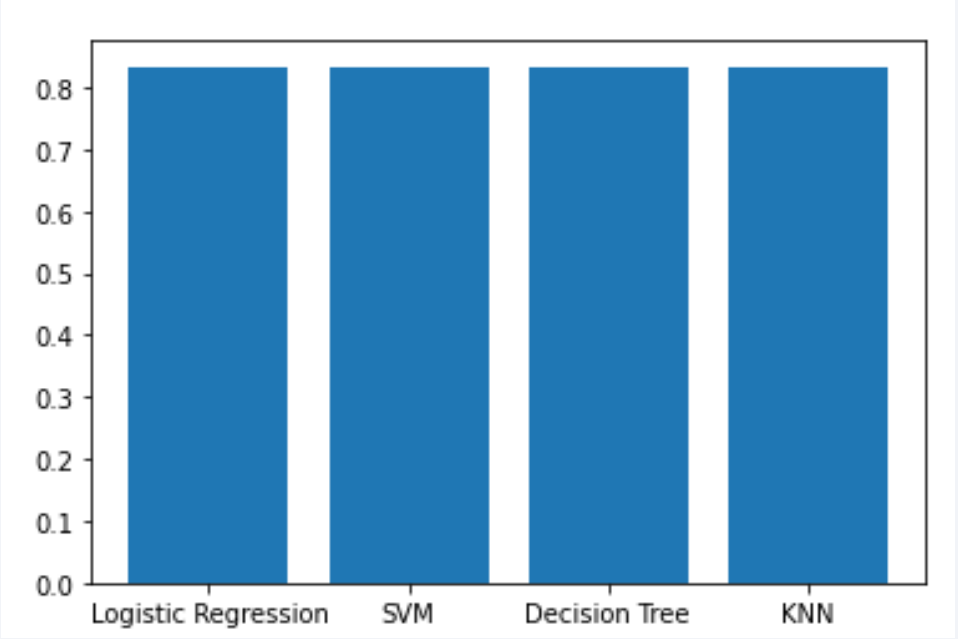


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- We could see from the following bar chart that all models have similar accuracy score which tell us that all models perform similarly



Confusion Matrix

- All models produce the same confusion matrix. From the confusion matrix we could see that the models can predict accurately the landed outcomes, but for the unsuccessful outcomes the models still couldn't predict the result accurately.

Conclusions

- We performed predictive analysis using logistic regression, support vector machine, decision tree, and K Nearest Neighbor
- All models performed similarly and have similar accuracy score

Appendix

https://github.com/faizamir123/IIBM_Capstone

Thank you!