IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ferris Wong
June,14th 2022

Outline

Executive Summary

- The Falcon 9 rocket launches with a cost of 62 million dollars according to SpaceX
- In this project, we will use the 4 classification models of Logistic Regression, Decision Tree, SVM and KNN to predict if the first stage will land
- Before we employ the models, we must clean, format, normalize, and visualize the data.
- After running our models, our accuracy is 83.33%, which is pretty good.

Introduction

- Space X 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 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 space X for a rocket launch.
- In this project, we will create a machine learning pipeline to predict if the first stage will land.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

Data Collection includes 3 main steps

1. Identify source of data
   - Data can come from a website, a file, an API, or a Database
   - By identifying where the source of the data is, will help us acquire it
2. Getting the Data
   - We used the Rest API in this project
3. Collect and Clean the Data
   - Collect the relevant information to analyze
   - Clean raw data

Data Collection – SpaceX API

- Collect Data from SpaceX API, from following Steps

1. Request to SpaceX API:
   - Using HTTP Requests to get data from SpaceX API
2. Parsing Data:
   - Using the GET request to parse the data and get the JSON request
   - Using the json_normalize method to convert the json result into a data frame
3. Collect and Clean Data:
   - Get info about the launches using the IDs given for each launch. Specifically, we will collect relevant columns to analyze
   - Clean raw data

Data Wrangling

1. Identify and handle the missing values
2. Data Formatting
   - Standardize the values into the same format, or unit, or convention
3. Data normalization
   - Bring all data into a similar range for more useful comparison
4. Data binning/turning
   - Binning creates bigger categories from a set of numerical values
   - Turning Categorical values to numeric variables to make statistical modeling easier

EDA with Data Visualization

- We use data visualization to explore the data and we used the following chart to do so:
  - Scatter Plot
  - Bar Chart
  - Line Chart
- These charts help us:
  - Visualize the relationship between the variables
  - Obtain some preliminary insight about how some variable would affect the success rate and we would use that information in future modules

EDA with SQL

- We performed some SQL Queries with these commands
  - Select
  - Sum()
  - Avg()
  - Min()
  - Max()
  - Count()
  - Sub query

Build an Interactive Map with Folium

- With Folium, we made an interactive map and added some map objects which include:
  - Markers
  - MarkerClusters
  - Circle
  - Line
- These objects are used to:
  - Mark all Launch sites on map
  - Mark the success/failed launches for each site on the map
  - Calculate the distances between a launch site to its proximities

Predictive Analysis (Classification)

- After cleaning the data, we did the following:
  - Split the data in training and test sets
  - With the Training set we perform cross validation
  - Then we tune the parameters
  - Find the best model
  - Finally with the test data, we evaluate the model
- We apply this process for all the classification models mentioned earlier in the report

Results

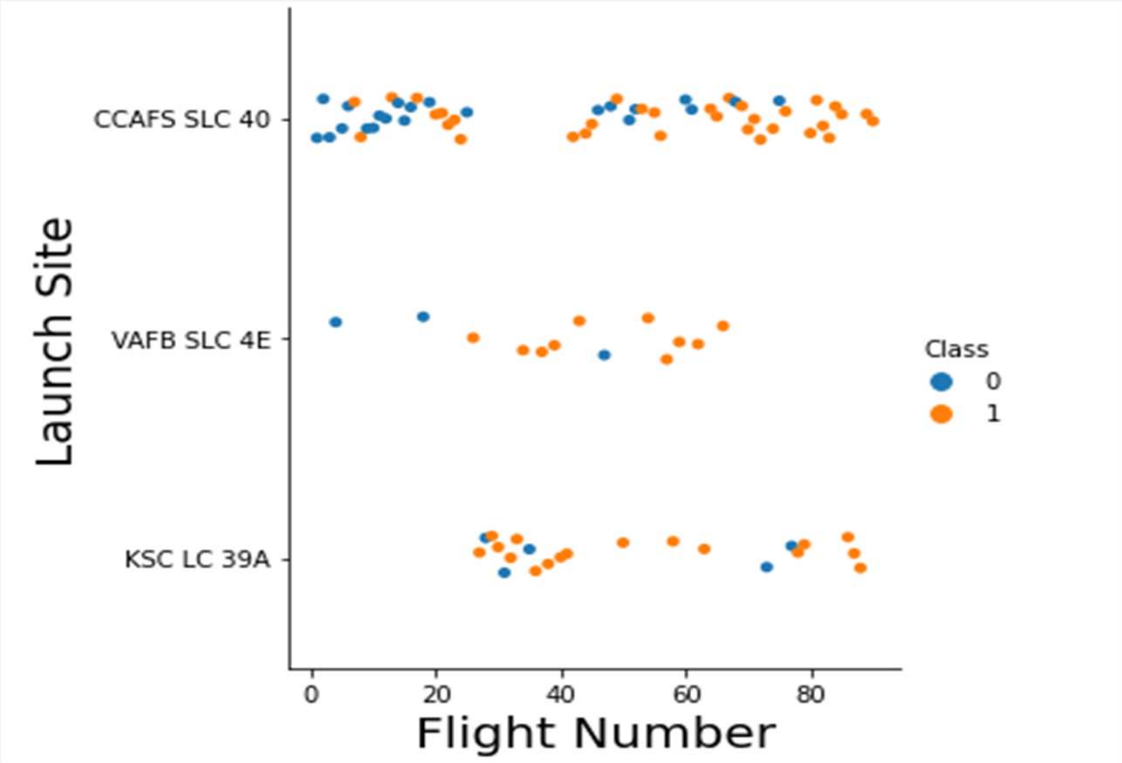
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

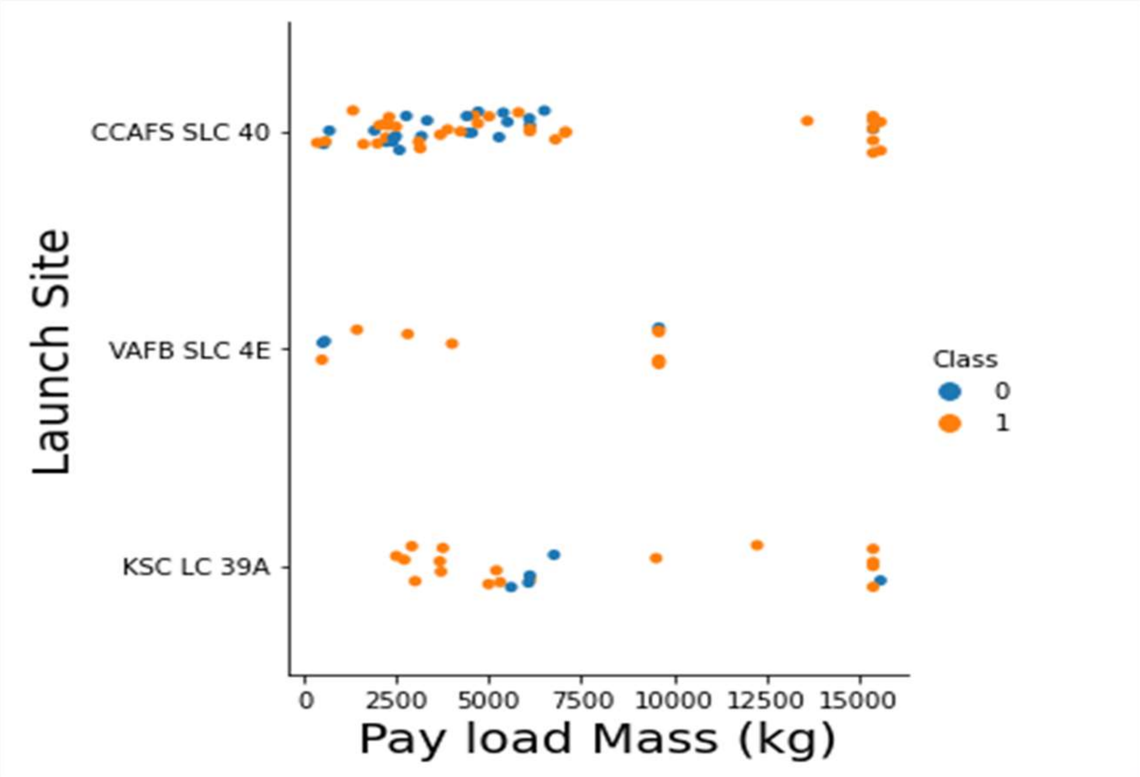
Flight Number vs. Launch Site

- We can see that:
  - The success rate increase when the flight number increases



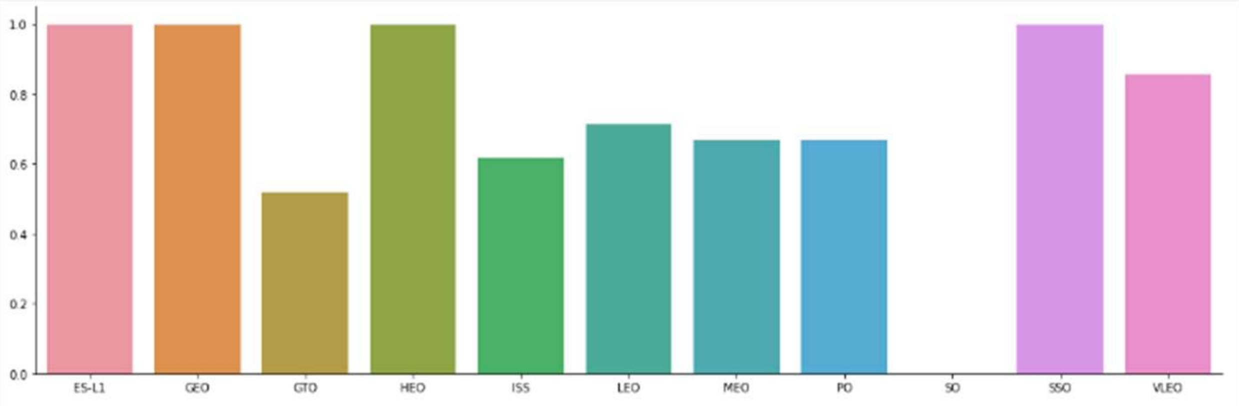
Payload vs. Launch Site

- From the graph shown we can conclude that:
  - The success rate is high when the payload mass is > 7000kg
  - We have a 100% success rate at KSC LC-39A when the payload mass is less than 5500kg



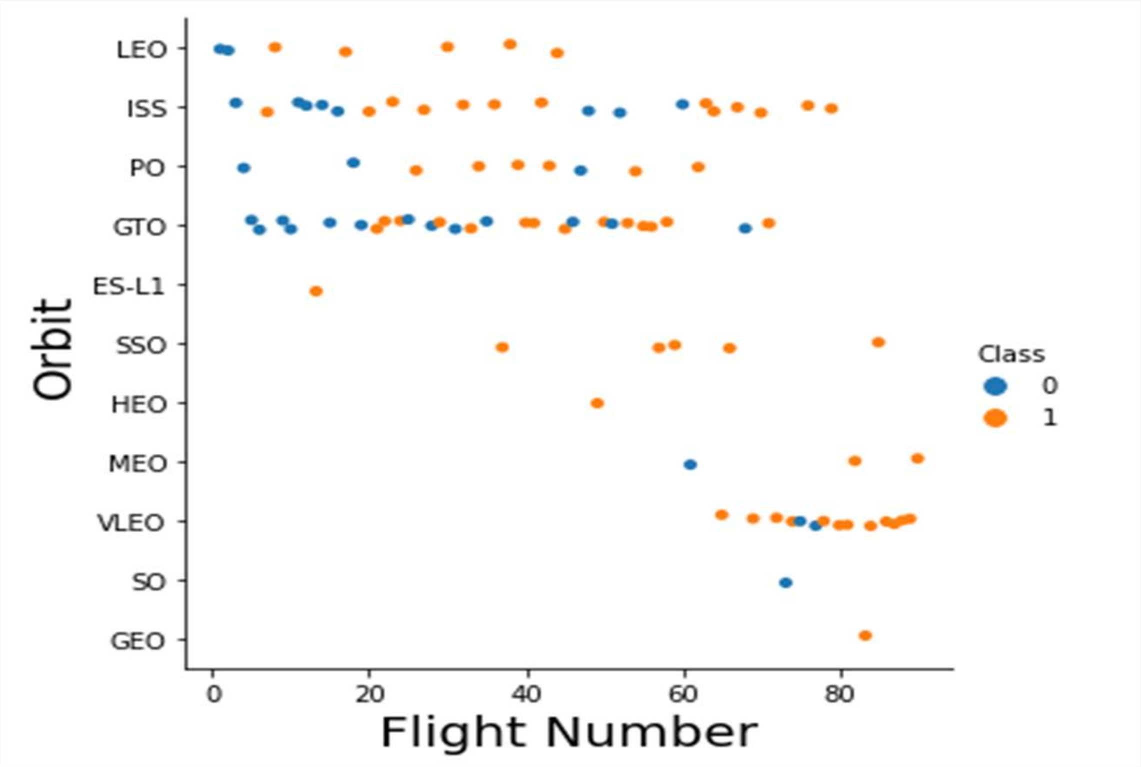
Success Rate vs. Orbit Type

- From the graph shown we can conclude that:
  - ES-L1, GEO, HEO, and SSO are the only orbit types which have a success rate of 100%
  - SO is the only orbit type to have a 0% success rate



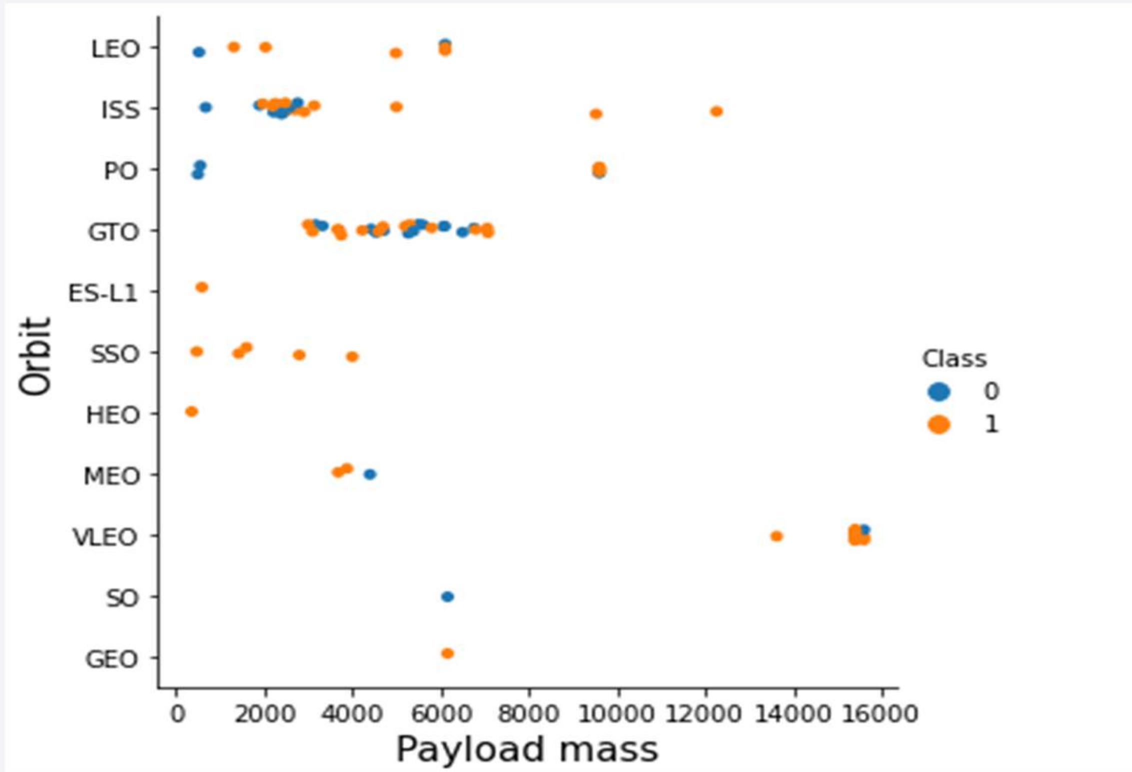
Flight Number vs. Orbit Type

- From the graph shown, we can conclude that:
  - When the flight number increases, the success rate increases too



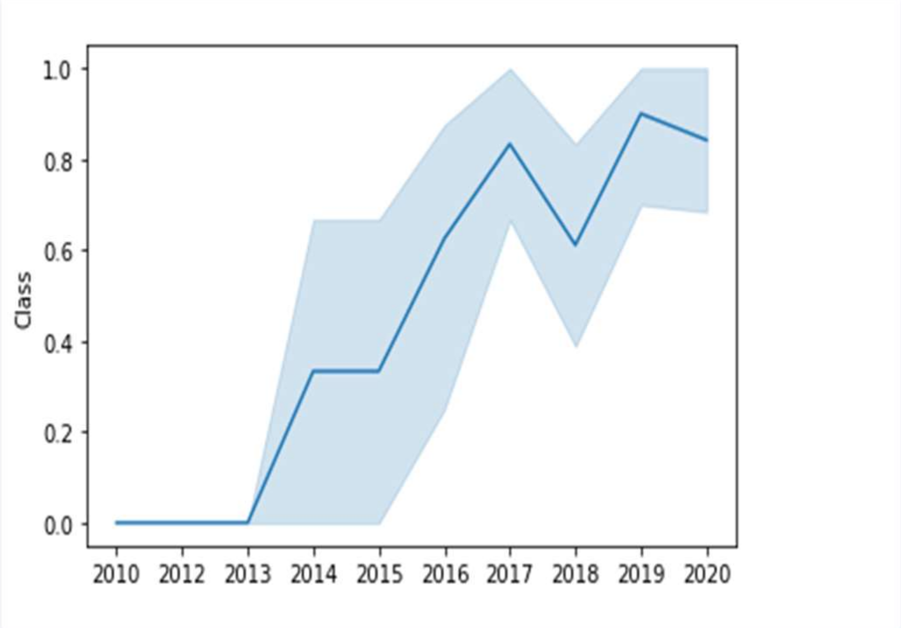
Payload vs. Orbit Type

- From the graph shown, we can conclude that:
  - The success rate is high when the payload mass is > 7000 kg



Launch Success Yearly Trend

- From the graph shown, we can conclude that:
  - There has been a steady increase in success rates since 2013 all the way to 2020



All Launch Site Names

- Find 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

- We used the ‘distinct’ query for getting the names of the launch sites

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
%sql select LAUNCH_SITE FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- We used the ‘distinct’ and ‘%’ to get the results

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL
```

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

sum(PAYLOAD_MASS__KG_)
619967

- We used the ‘sum()’ functions and ‘like’ operator

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL
```

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

AVG(PAYLOAD_MASS__KG_)
6138.287128712871

- We used the 'avg()' function to get the average payload mass

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
%sql select min(DATE) from SPACEXTBL;
```

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

min(DATE)
01-03-2013

- To find the first date where the landing outcome was successful, we used the ‘Min(Date)’ function

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%sql select payload from spacextbl where landing__outcome like 'Success (drone ship)' and payload_mass__k
```

```
 * ibm_db_sa://yhh17832:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomai
n.cloud:32731/bludb
Done.
```

payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

- Select distinct Booster_Version from SPACEXDATASET where (Landing Outcome = 'Success (drone ship)') and (PAYLOAD_MASS KG_ > 4000 and PAYLOAD_MASS KG_ < 6000)

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

missionoutcomes
1
98
1
1

- Select count(*) as "Total number", Mission_Outcome from SPACEXDATASET where (Mission_Outcome LIKE '%Success%') or (Mission_Outcome LIKE '%Failure%') group by Mission_Outcome order by "Total number" desc, Mission_Outcome desc

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_
```

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

boosterversion
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

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

```
%sql select booster_version,launch_site from spacextbl where landing__outcome not like 'Success (drone sh
```

```
 * ibm_db_sa://yhh17832:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomai
n.cloud:32731/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1013	CCAFS LC-40
F9 v1.1 B1014	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40
F9 v1.1 B1016	CCAFS LC-40
F9 v1.1 B1018	CCAFS LC-40
F9 FT B1019	CCAFS LC-40

- SELECT MONTHNAME(Date) as "Month Name", Landing Outcome, Booster_Version, Launch_Site FROM SPACEXDATASET WHERE (YEAR(Date) = '2015') and (Landing Outcome LIKE '%Failure%') ORDER BY Month(Date) ASC

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

- 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 FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DAT
```

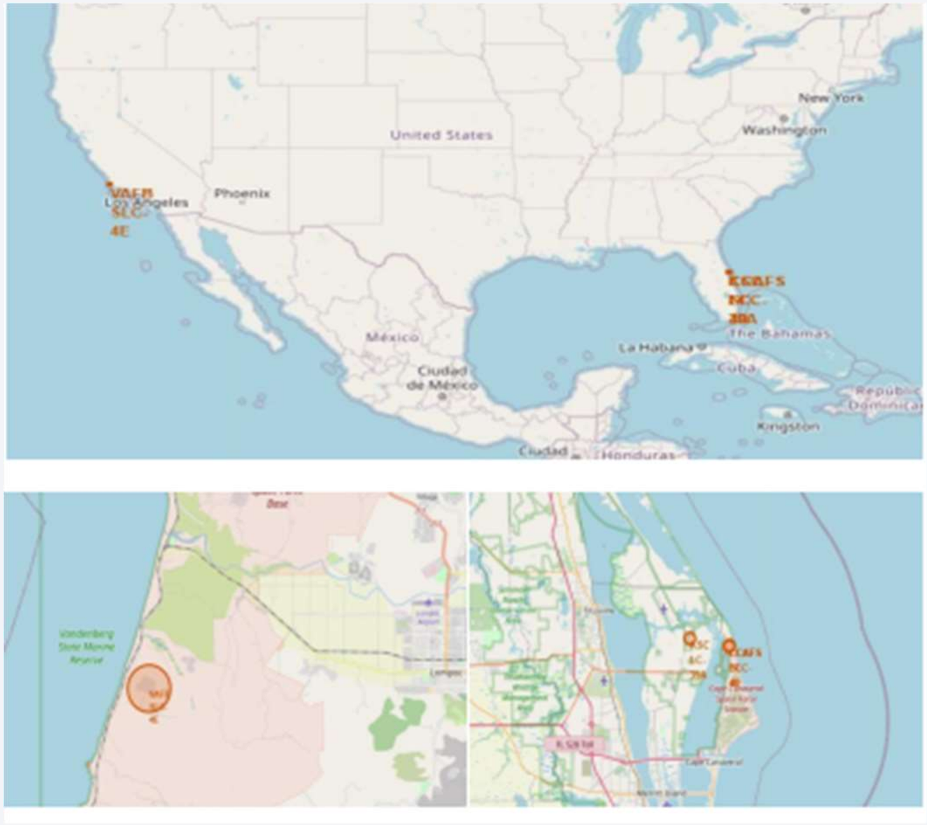
- SELECT COUNT(*) AS "COUNT", Landing Outcome FROM SPACEXDATASET WHERE (Date BETWEEN '2010-06-04' AND '2017-03-20') AND (Landing Outcome LIKE '%Success%') GROUP BY Landing__Outcome ORDER BY "COUNT" DESC

Section 3

Launch Sites Proximities Analysis

Launch Sites

- Launch Sites:
  - Near the equator
  - Near the beach
  - Far from residential area

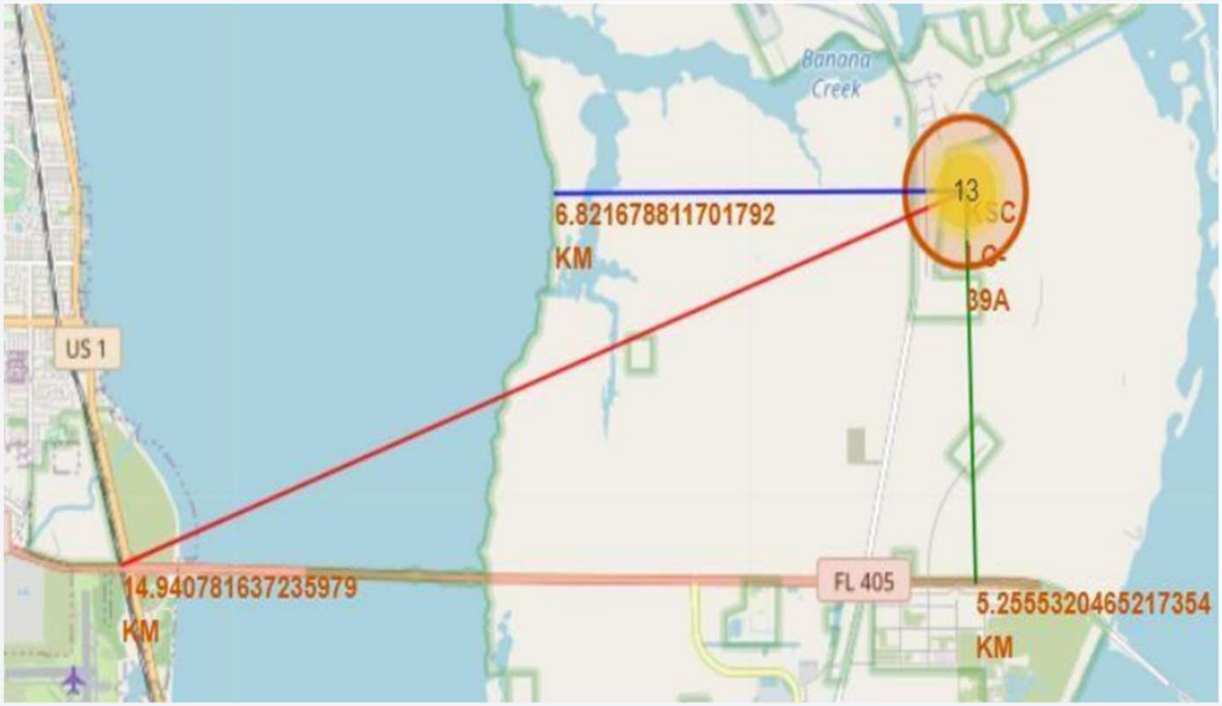


Successful and Failed launches in each site

	Total Launches	Successful Launches	Failed Launches	% Success Rate
KSC LC-39A	13	10	3	76.9%
CCAFS SLC-40	7	3	4	42.9%
VAFB SLC-4E	10	4	6	40%
CCAFS LC-40	26	7	19	26.9%

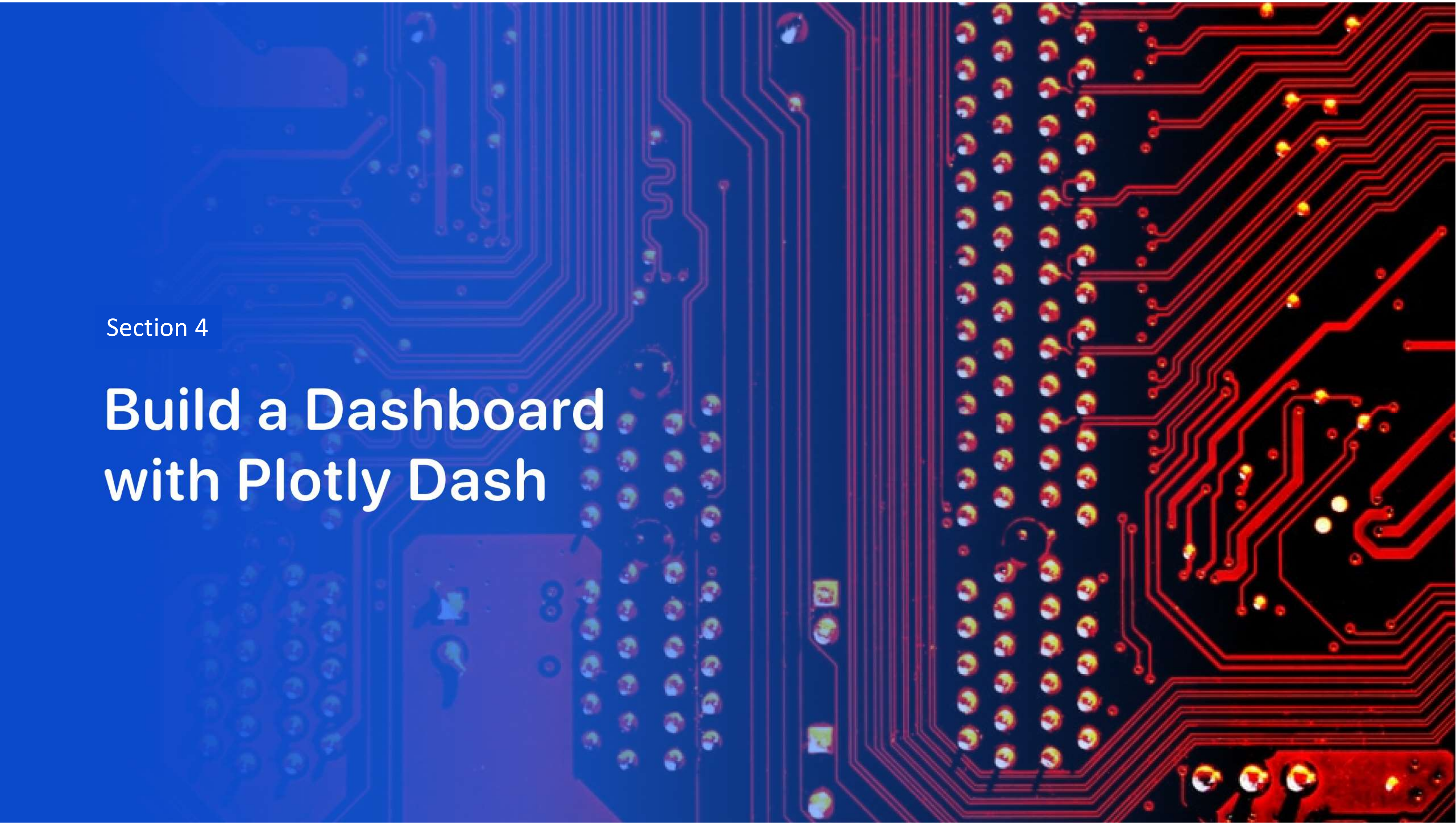
Distance from Site to Proximity

- The Site launch is:
  - Near the beach
  - Far from residential area



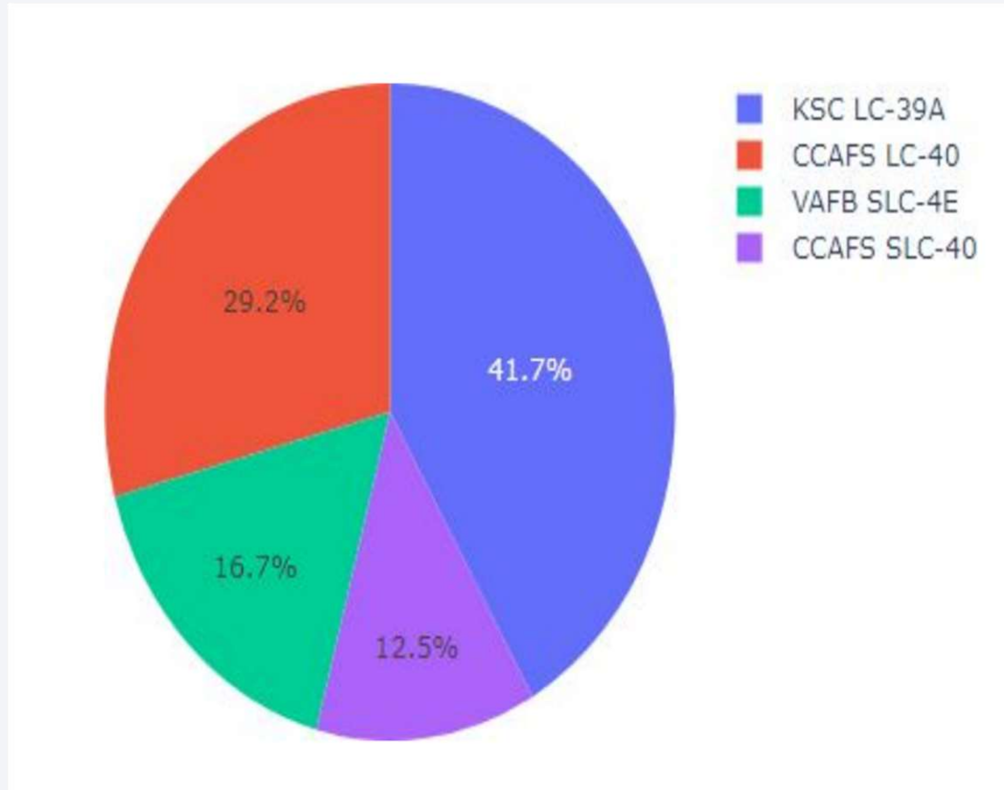
Section 4

Build a Dashboard with Plotly Dash



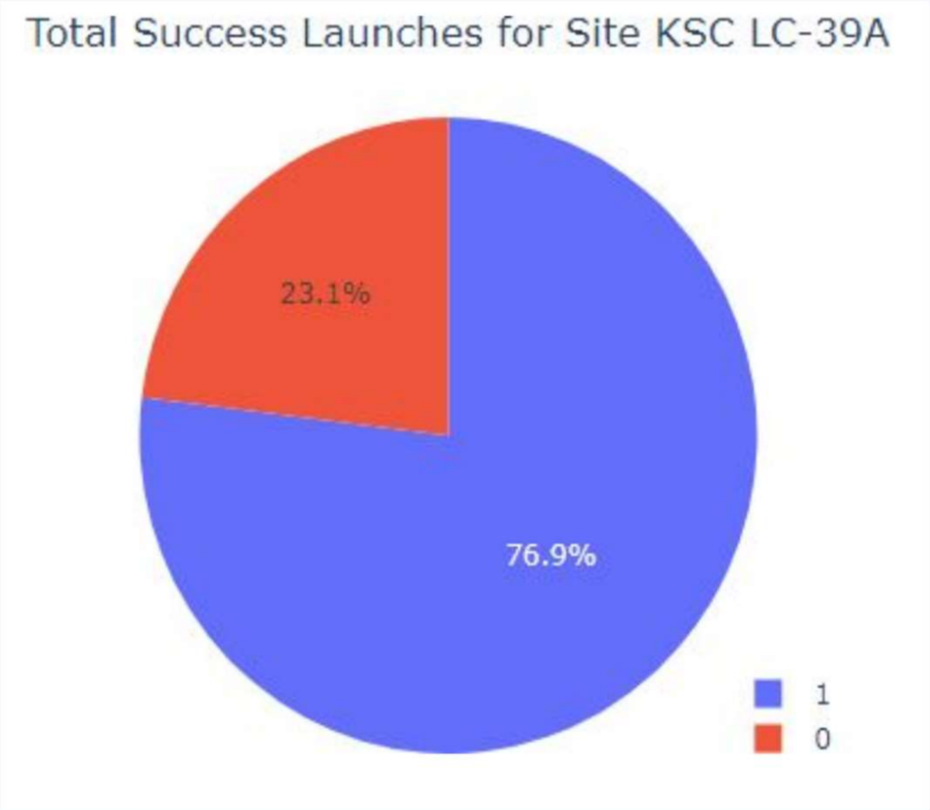
Successful launches from all sites

- Gatherings
  - KSC LC-39A is the site with the most successful launches
  - CCAFS SLC-40 has the least number of successful launches



Highest Launch Success Ratio

- KSC LC-39A is the site with the highest success ratio

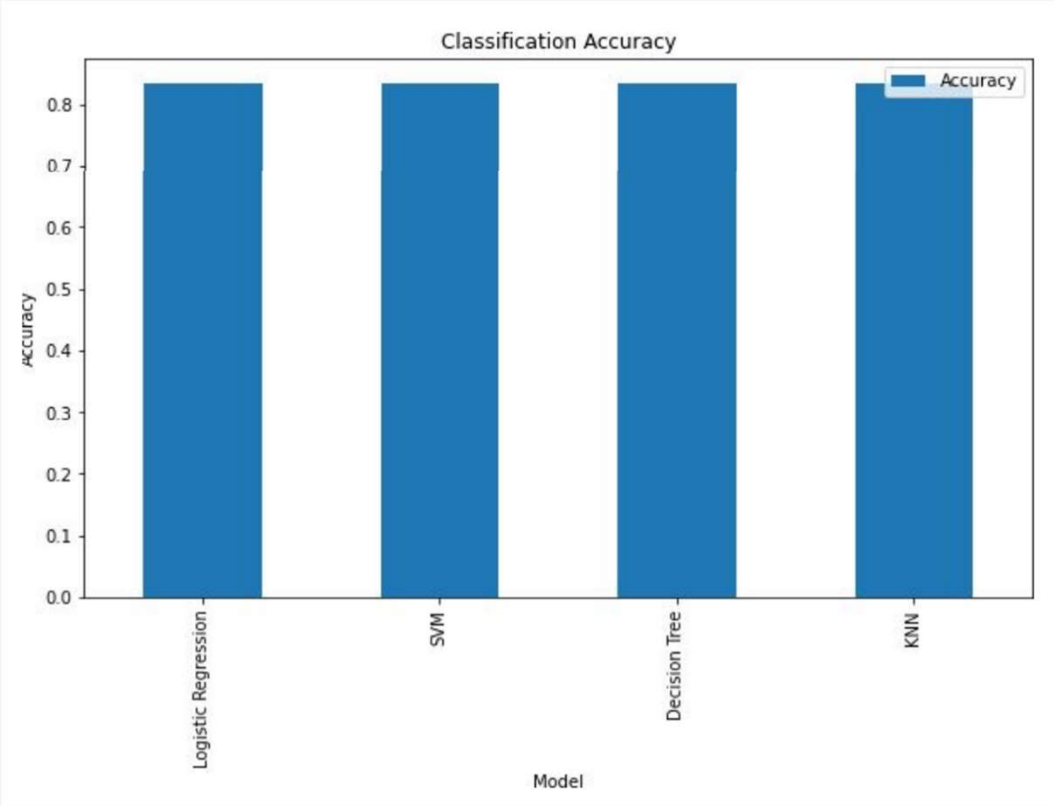


Section 5

Predictive Analysis (Classification)

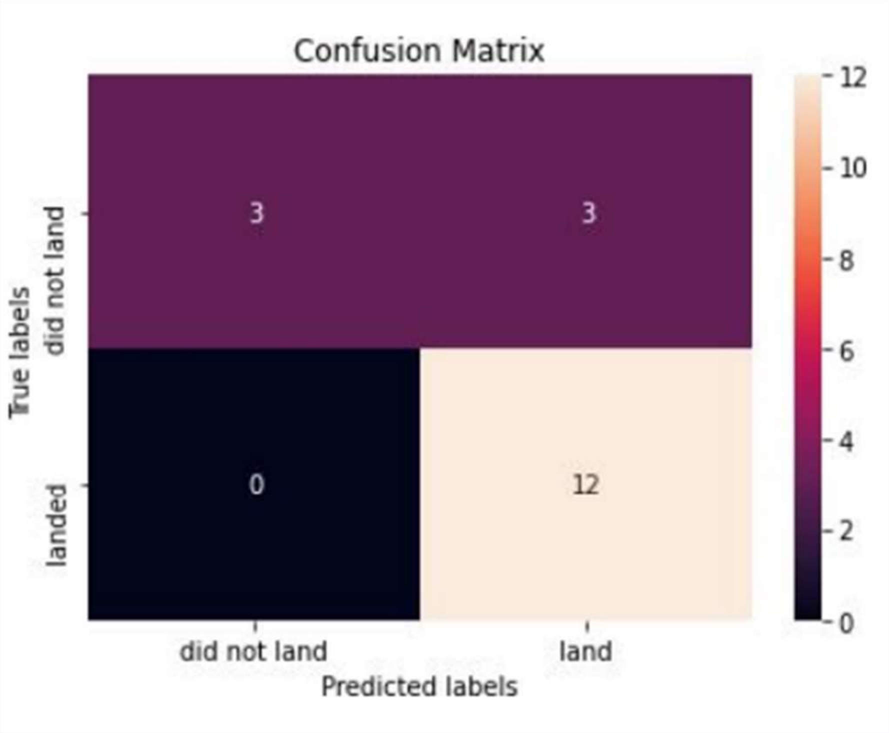
Classification Accuracy

- All models have the same accuracy of .833
- I predict that this is because the data set used to build and evaluate the model is too small



Confusion Matrix

- Since all models have the same accuracy, they have the same confusion matrix
- The confusion matrix itself is pretty good when the number of true positives is high and true negatives is low



Conclusions

- In this project, we created four classification models of Logistic regression, Decision Tree, SVM and KNN to predict if the first stage will land. And the accuracy of them is same 83.33%.
- We gained also some insights such as:
  - Launch success yearly trend increases when the number of flights increases.
  - KSC LC-39A site has the largest successful launches and the highest launch success rate.
  - Payload range from 2000 to 4000kg has the highest launch success rate, and above 5500kg has lowest launch success rate.
  - F9 Booster version of FT has the highest launch success rate.
  - ES-L1, GEO, HEO, SSO are the orbit types which have the success rate of 100%. On the contrary, SO orbit has not any successfully launches
- The results that we get are very positive, but in fact we have analyzed on a small data set. So, in the future when the data of launches is more, we will have a more complete


Thank you!