IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Rodolfo Rodrigues>
<03/22/2022>

Outline

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

Executive Summary

- Data were collected from web sources;
- Many pre-processing techniques were developed;
- In order to get a better data understanding EDA, map visualizations and SQL queries were developed;
- Some algorithms were trained and evaluated and compared;
- A classification model with a good accuracy (~84%) was build.

Introduction

- Since June 2010, rockets from the Falcon 9 family have been launched 148 times, with 146 full mission successes, one partial failure and one total loss of the spacecraft.
- The Falcon design features reusable first-stage boosters, which land either on a ground pad near the launch site or on a drone ship at sea.
- 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 information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- This project's goal is to predict the landing outcome (success or fail) for future rocket launching

Section 1

Methodology

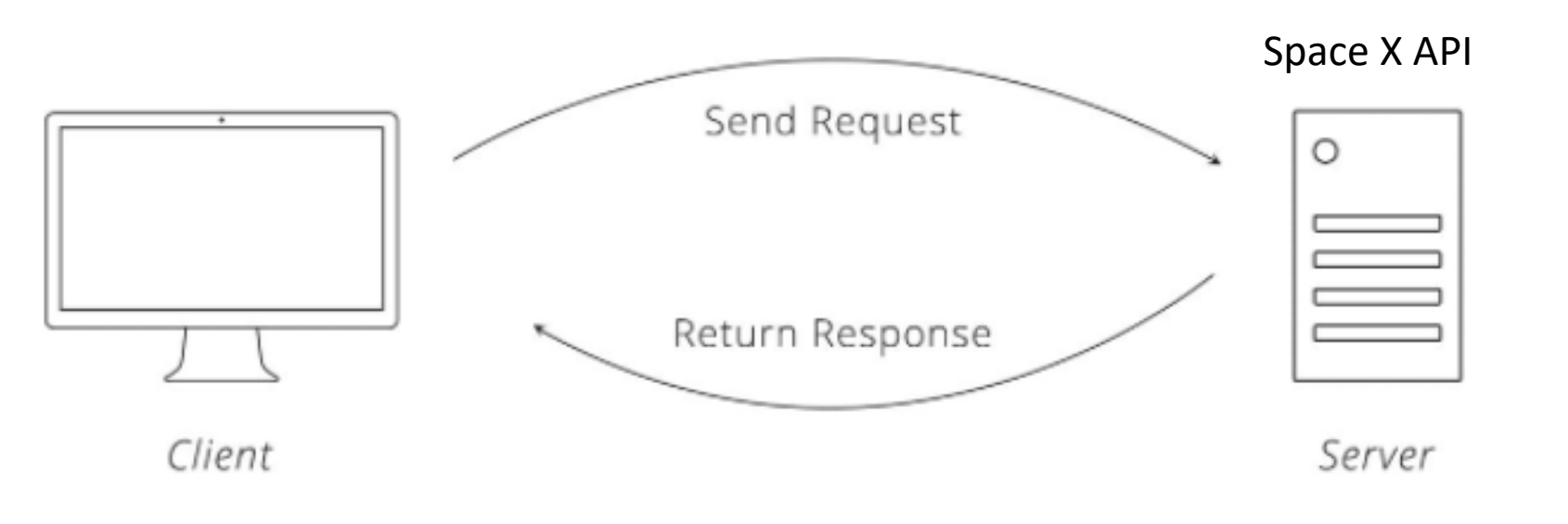
Methodology

Executive Summary

- Data collection methodology:
  - Data were collected through SpaceX API and by webscrapping
- Perform data wrangling
  - Many pre-processing techniques were developed to get a confident 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
  - Models were trained and fine tunned on train set (80% of dataset) and evaluated on test set.

Data Collection

- Data were collected from web sources mainly SpaceX API. Wikipedia page webscrapping were also useful.
- Request Space X API



Data Collection

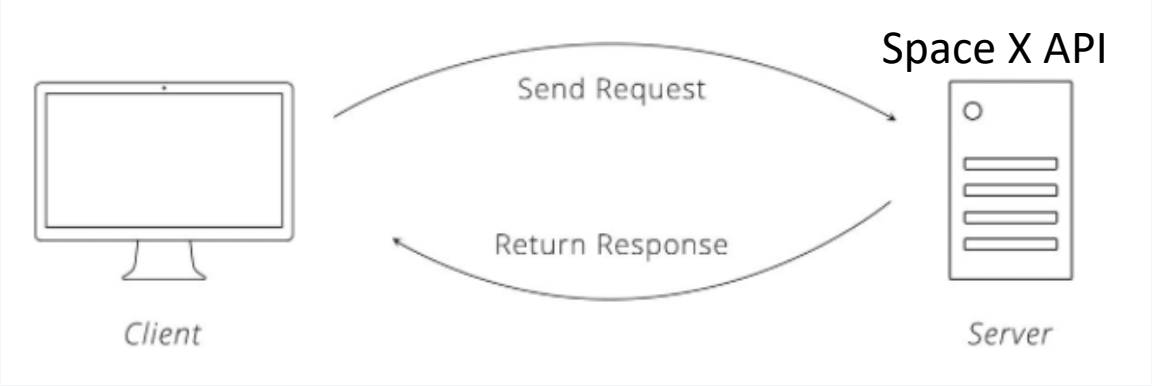
- Data were collected from web sources mainly SpaceX API. Wikipedia page webscrapping were also useful.
- Wikipidia scrapping

2010 to 2013

[hide] Flight No.	Date and time (UTC)	Version, Booster [b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
1	4 June 2010, 18:45	F9 v1.0[7] B0003.1[8]	CCAFS, SLC-40	Dragon Spacecraft Qualification Unit		LEO	SpaceX	Success	Failure[9][10] (parachute)
	First flight of Falcon 9 v1.0.[11] Used a boilerplate version of Dragon capsule which was not designed to separate from the second stage.(more details below) Attempted to recover the first stage by parachuting it into the ocean, but it burned up on reentry, before the parachutes even deployed.[12]								

Data Collection – SpaceX API

- Data were collected by:
- requests.get(spacex_url)
- Some functions were defined the get the data: getLaunchSite(data), getBoosterVersion(data) etc
- Request and parse the SpaceX launch data using the GET request
- Filter the dataframe to only include Falcon 9 launches




```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False

https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/1-1-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Web scraping Falcon 9 and Falcon Heavy Launches Records from Wikipedia. More specifically, the launch records are stored in a HTML table.

1. Request the Falcon9 Launch Wiki page from its URL
2. Extract all column/variable names from the HTML table header
3. Create a data frame by parsing the launch HTML tables

2020 [edit]

In late 2019, Gwynne Shotwell stated that SpaceX hoped for as many as 24 launches for Starlink satellites in 2020,[490] in addition to 14 or 15 non-Starlink launches. At 26 launches, 13 of which for Starlink satellites, Falcon 9 had its most prolific year, and Falcon rockets were second most prolific rocket family of 2020, only behind China's Long March rocket family.[491]

[hide] Flight No.	Date and time (UTC)	Version, Booster[b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
78	7 January 2020, 02:19:21[492]	F9 B5 △ B1049.4	CCAFS, SLC-40	Starlink 2 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third large batch and second operational flight of Starlink constellation. One of the 60 satellites included a test coating to make the satellite less reflective, and thus less likely to interfere with ground-based astronomical observations.[493]								
79	19 January 2020, 15:30[494]	F9 B5 △ B1046.4	KSC, LC-39A	Crew Dragon in-flight abort test[495] (Dragon C205.1)	12,050 kg (26,570 lb)	Sub-orbital[496]	NASA (CTS)[497]	Success	No attempt
	An atmospheric test of the Dragon 2 abort system after Max Q. The capsule fired its SuperDraco engines, reached an apogee of 40 km (25 mi), deployed parachutes after reentry, and splashed down in the ocean 31 km (19 mi) downrange from the launch site. The test was previously slated to be accomplished with the Crew Dragon Demo-1 capsule;[498] but that test article exploded during a ground test of SuperDraco engines on 20 April 2019.[419] The abort test used the capsule originally intended for the first crewed flight.[499] As expected, the booster was destroyed by aerodynamic forces after the capsule aborted.[500] First flight of a Falcon 9 with only one functional stage – the second stage had a mass simulator in place of its engine.								
80	29 January 2020, 14:07[501]	F9 B5 △ B1051.3	CCAFS, SLC-40	Starlink 3 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third operational and fourth large batch of Starlink satellites, deployed in a circular 290 km (180 mi) orbit. One of the fairing halves was caught, while the other was fished out of the ocean.[502]								
81	17 February 2020, 15:05[503]	F9 B5 △ B1056.4	CCAFS, SLC-40	Starlink 4 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Failure (drone ship)
	Fourth operational and fifth large batch of Starlink satellites. Used a new flight profile which deployed into a 212 km × 386 km (132 mi × 240 mi) elliptical orbit instead of launching into a circular orbit and firing the second stage engine twice. The first stage booster failed to land on the drone ship[504] due to incorrect wind data.[505] This was the first time a flight proven booster failed to land.								

https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/1-2-webscraping.ipynb

Data Wrangling

- We performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models

1. Calculate the number of launches on each site
2. Calculate the number and occurrence of each orbit
3. Calculate the number and occurence of mission outcome per orbit type
4. Create a landing outcome label from Outcome column

Class	FlightNumber
0	30
1	60

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

We used the following line of code to determine the success rate:

```
df["Class"].mean()
```

```
0.6666666666666666
```

https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/1-3-data-wrangling.ipynb

EDA with Data Visualization

- Exploratory Data Analysis were performed in order to get a better data understanding. Main plots created:
- The relationship between Flight Number and Launch Site;
- The relationship between Payload and Launch Site;
- The relationship between success rate of each orbit type;
- The relationship between Flight Number and Orbit type;
- The relationship between Payload and Orbit type;
- The launch success yearly trend;

https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/2-2-eda-dataviz.ipynb

EDA with SQL

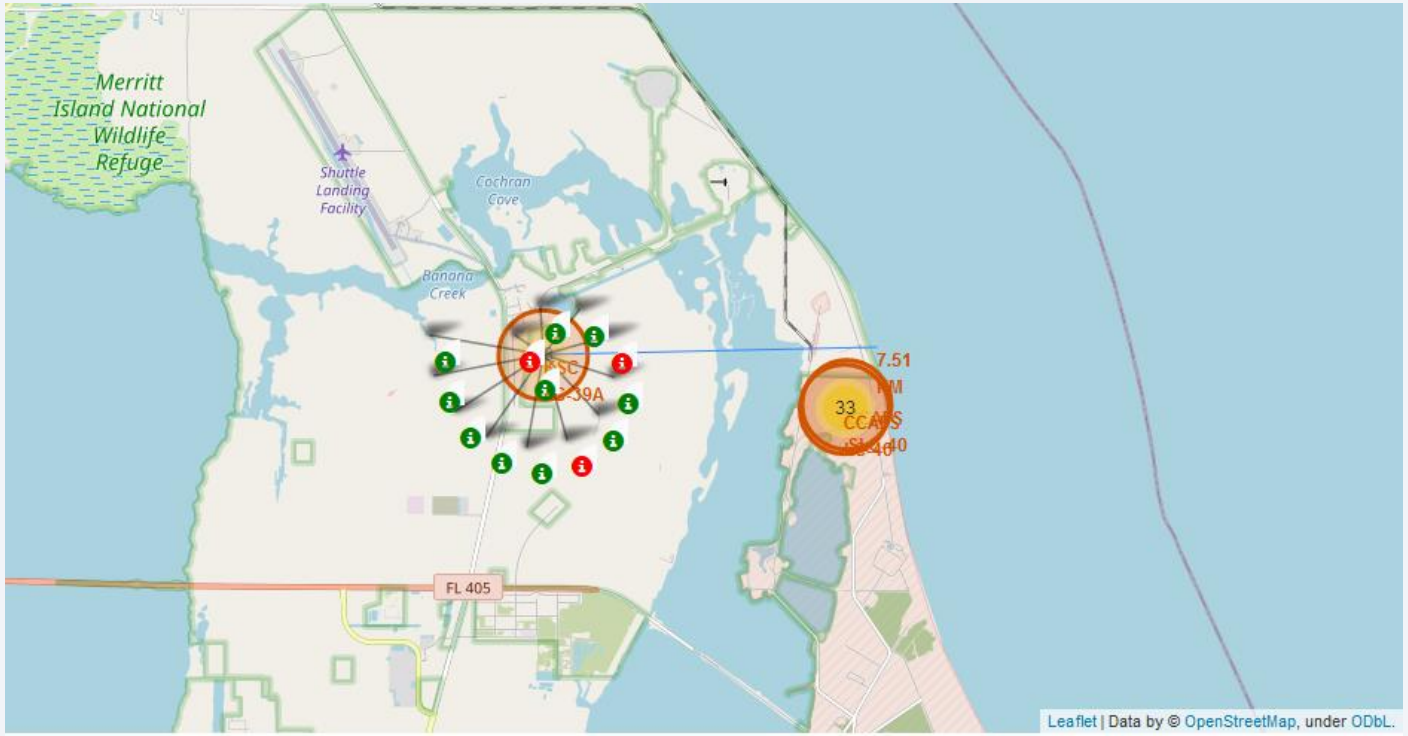
Many SQL queries were performed in order the get a better understanding of data. Some of them include:

- Display the total payload mass carried by boosters launched by NASA (CRS);
- Display average payload mass carried by booster version F9 v1.1;
- List 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/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/2-1-eda-sql.ipynb

Interactive Map with Folium

We build some interactive objects such as markers, circles, lines, etc. We created them and added to a folium map to get a better data understanding.



https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/3-1-launch-site-location.ipynb

Predictive Analysis (Classification)

- Dataset was standardized;
- Dataset was splitted in train and test set (20%);
- Logistic Regression, SVM, KNN, Decision Tree algorithms were fitted using cross validation (10-folds) technique. Fine tunning was performed.
- Some model comparisons were done.

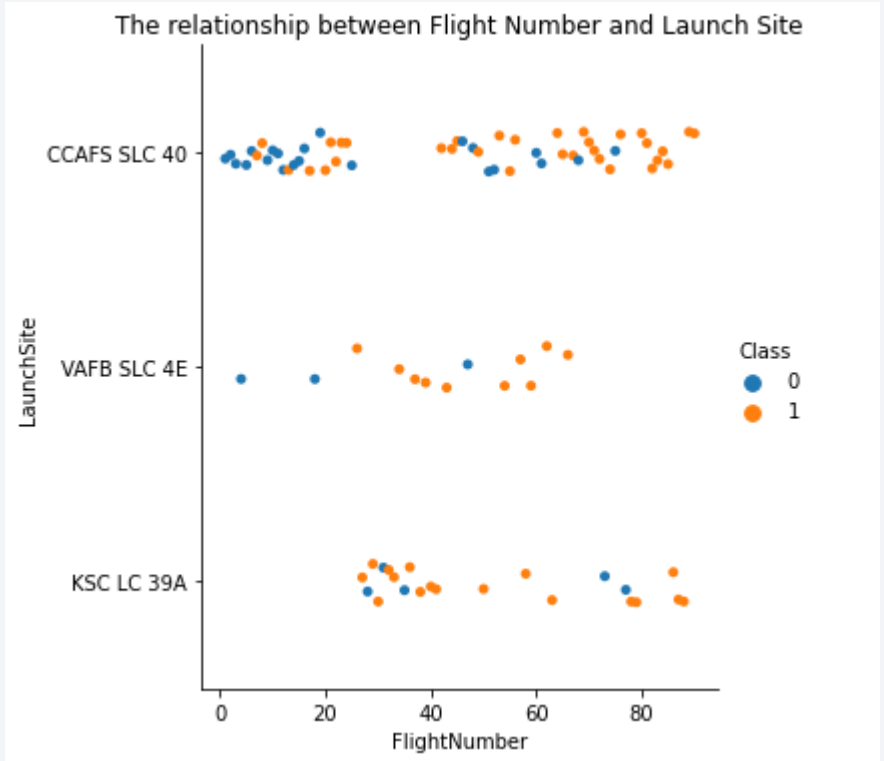
https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/4-1-spacex-machine-learning-prediction.ipynb

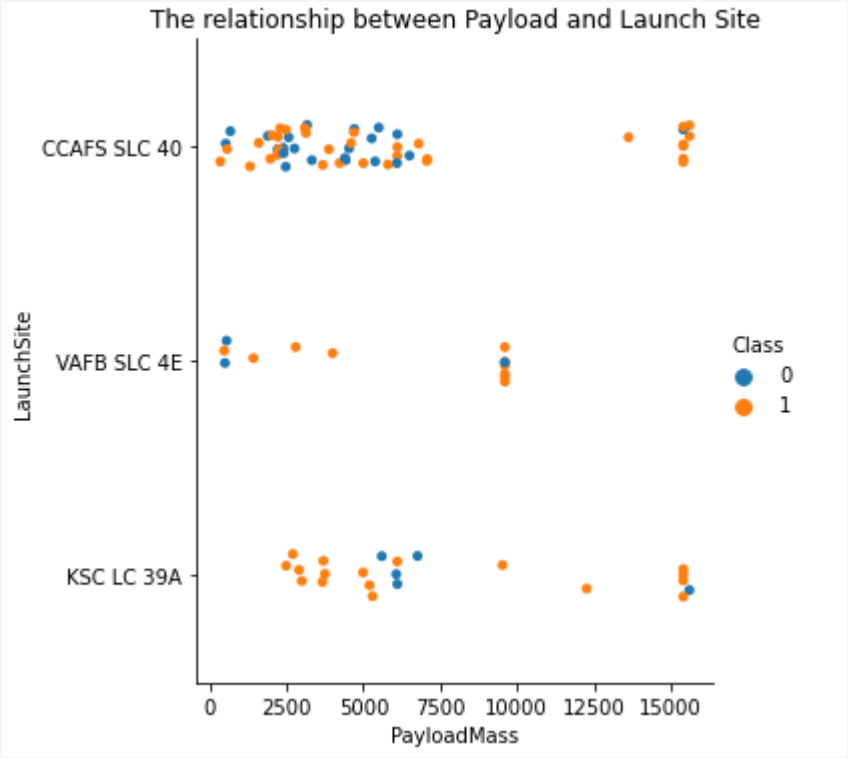
Section 2

Insights drawn from EDA

EDA with Data Visualization

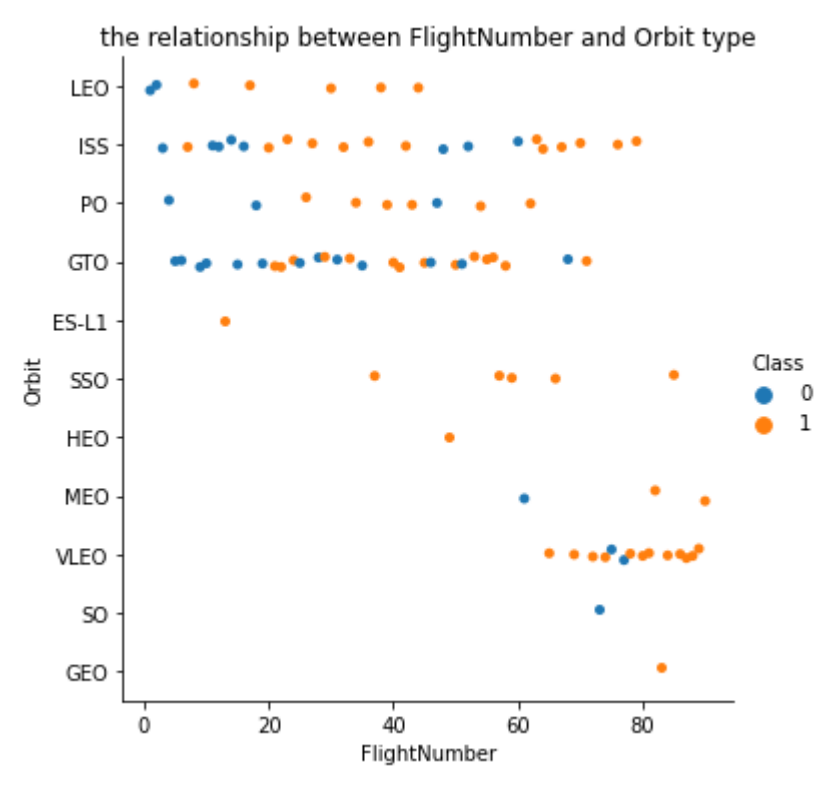
We performed some Exploratory Data Analysis (EDA) to find some patterns and find out how the target class is related to the features.

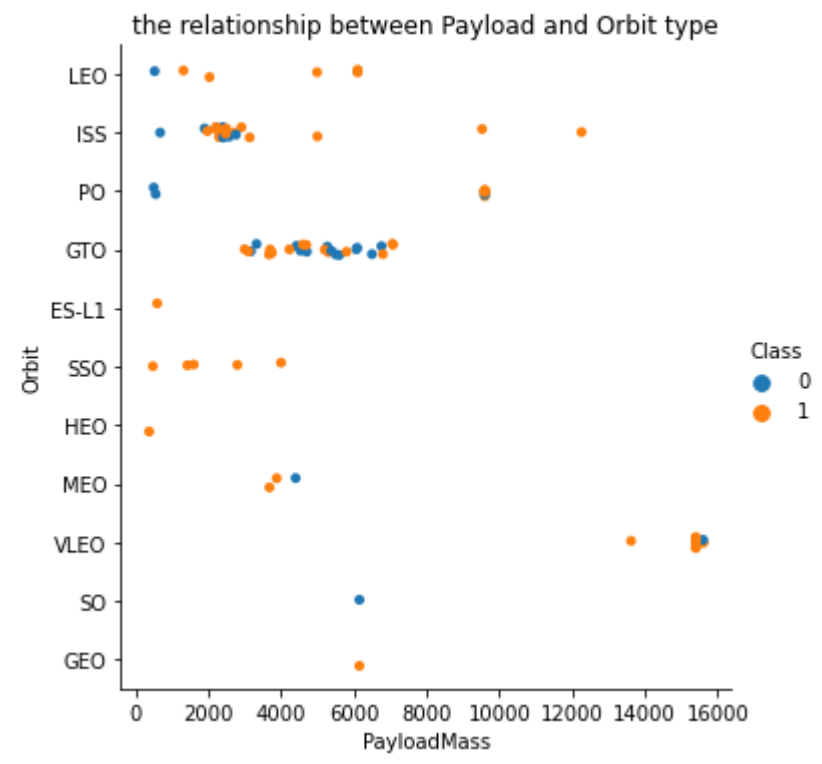




EDA with Data Visualization

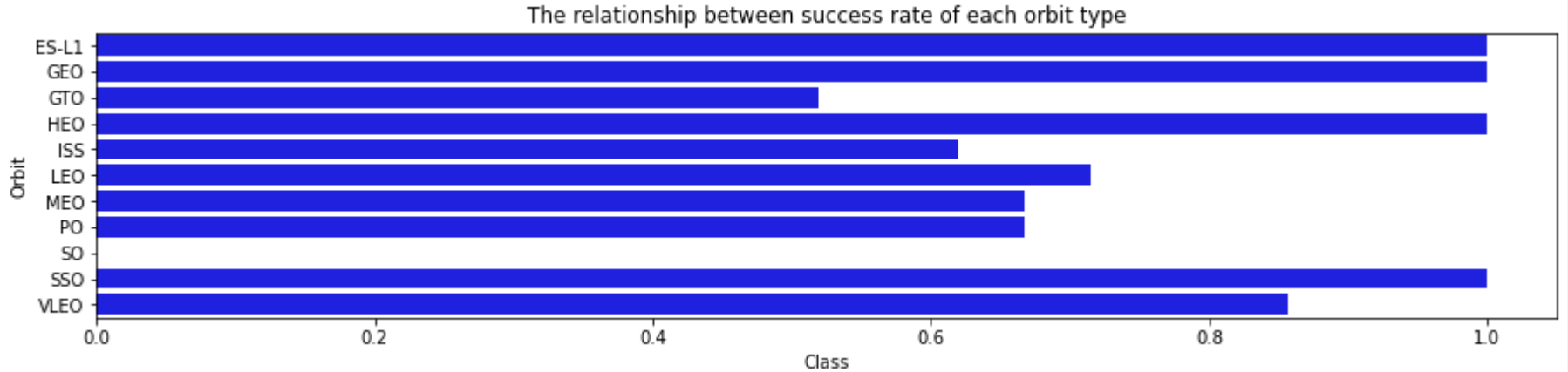
We performed some Exploratory Data Analysis (EDA) to find some patterns and fing out how the target class is related to the features.



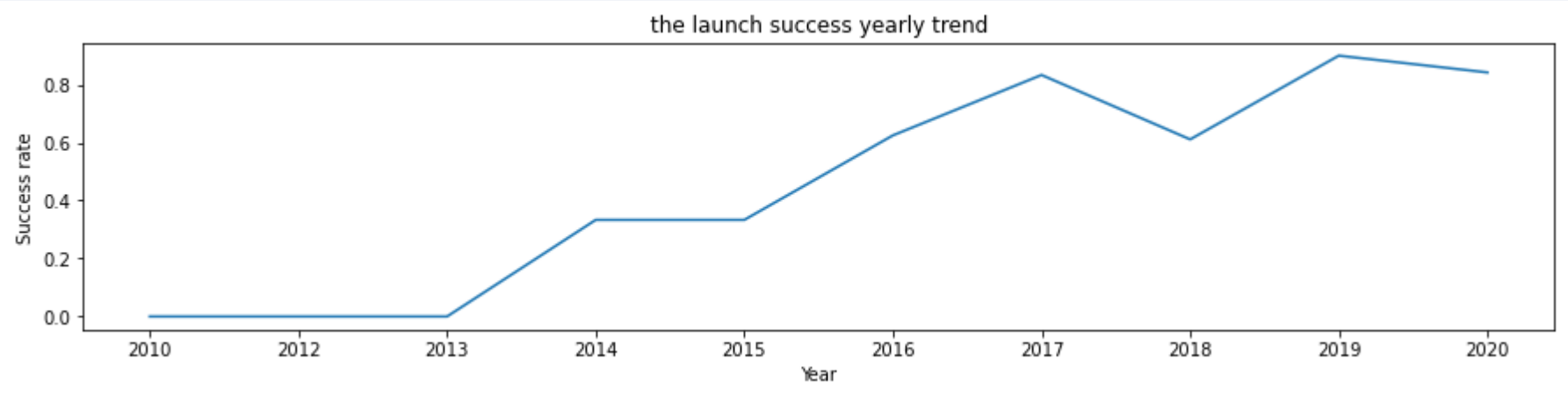


EDA with Data Visualization

Orbits that have high success rate



The success rate since 2013 kept increasing until 2020



All Launch Site Names

- Find the names of the unique launch sites

```
selectQuery = "SELECT DISTINCT(Launch_Site) FROM spacexdataset"
pandas.read_sql(selectQuery, pconn)
```

	LAUNCH_SITE
0	CCAFS LC-40
1	CCAFS SLC-40
2	KSC LC-39A
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

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

```
selectQuery = "SELECT * FROM spacexdataset WHERE Launch_Site like 'CCA%' LIMIT 5"
pandas.read_sql(selectQuery, pconn)
```

	DATE	TIME__UTC_	BOOSTER_VERSION	LAUNCH_SITE	PAYLOAD	PAYLOAD_MASS__KG_	ORBIT	CUSTOMER	MISSION_OUTCOME	LANDING__OUTCOME
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
selectQuery = "SELECT SUM(PAYLOAD_MASS__KG_) AS NASA_PAYLOAD from spacexdataset WHERE Customer = 'NASA (CRS)';"
pandas.read_sql(selectQuery, pconn)
```

	NASA_PAYLOAD
0	45596

Average Payload Mass by F9 v1.1

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

```
selectQuery = "SELECT AVG(PAYLOAD_MASS__KG_) as AVG_PAYLOAD FROM spacexdataset WHERE Booster_Version = 'F9 v1.1';"
pandas.read_sql(selectQuery, pconn)
```

	AVG_PAYLOAD
0	2928

First Successful Ground Landing Date

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

```
selectQuery = "SELECT min(Date) as DATE FROM spacexdataset  WHERE LANDING__OUTCOME = 'Success (ground pad)';"
pandas.read_sql(selectQuery, pconn)
```

	DATE
0	2015-12-22

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

```
selectQuery = "SELECT DISTINCT(Booster_Version), PAYLOAD_MASS__KG_ FROM spacexdataset
WHERE Landing__Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;  "
pandas.read_sql(selectQuery, pconn)
```

	BOOSTER_VERSION	PAYLOAD_MASS__KG_
0	F9 FT B1021.2	5300
1	F9 FT B1031.2	5200
2	F9 FT B1022	4696
3	F9 FT B1026	4600

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
selectQuery = "SELECT count(*) as Sucess FROM spacexdataset WHERE Landing__Outcome like 'Success%';"
pandas.read_sql(selectQuery, pconn)
```

	SUCESS
0	61

```
selectQuery = "SELECT count(*) as FAIL FROM spacexdataset WHERE Landing__Outcome like 'Fai%';"
pandas.read_sql(selectQuery, pconn)
```

	FAIL
0	10

Boosters Carried Maximum Payload

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

```
selectQuery = "SELECT DISTINCT(Booster_Version) FROM spacexdataset
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM spacexdataset);"
pandas.read_sql(selectQuery, pconn)
```

	BOOSTER_VERSION
0	F9 B5 B1048.4
1	F9 B5 B1048.5
2	F9 B5 B1049.4
3	F9 B5 B1049.5
4	F9 B5 B1049.7
5	F9 B5 B1051.3
6	F9 B5 B1051.4
7	F9 B5 B1051.6
8	F9 B5 B1056.4
9	F9 B5 B1058.3
10	F9 B5 B1060.2
11	F9 B5 B1060.3

2015 Launch Records

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

```
selectQuery = "SELECT  DATE, LAUNCH_SITE, LANDING__OUTCOME, BOOSTER_VERSION FROM Spacexdataset
WHERE YEAR(DATE) = 2015 AND LANDING__OUTCOME = 'Failure (drone ship)'"
pandas.read_sql(selectQuery, pconn)
```

	DATE	LAUNCH_SITE	LANDING__OUTCOME	BOOSTER_VERSION
0	2015-01-10	CCAFS LC-40	Failure (drone ship)	F9 v1.1 B1012
1	2015-04-14	CCAFS LC-40	Failure (drone ship)	F9 v1.1 B1015

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

```
selectQuery = "SELECT LANDING__OUTCOME AS OUTCOME, COUNT(*) AS TOTAL FROM SPACEXDATASET
WHERE DATE BETWEEN '04/06/2010' AND '20/03/2017' GROUP BY LANDING__OUTCOME ORDER BY TOTAL DESC"
pandas.read_sql(selectQuery, pconn)
```

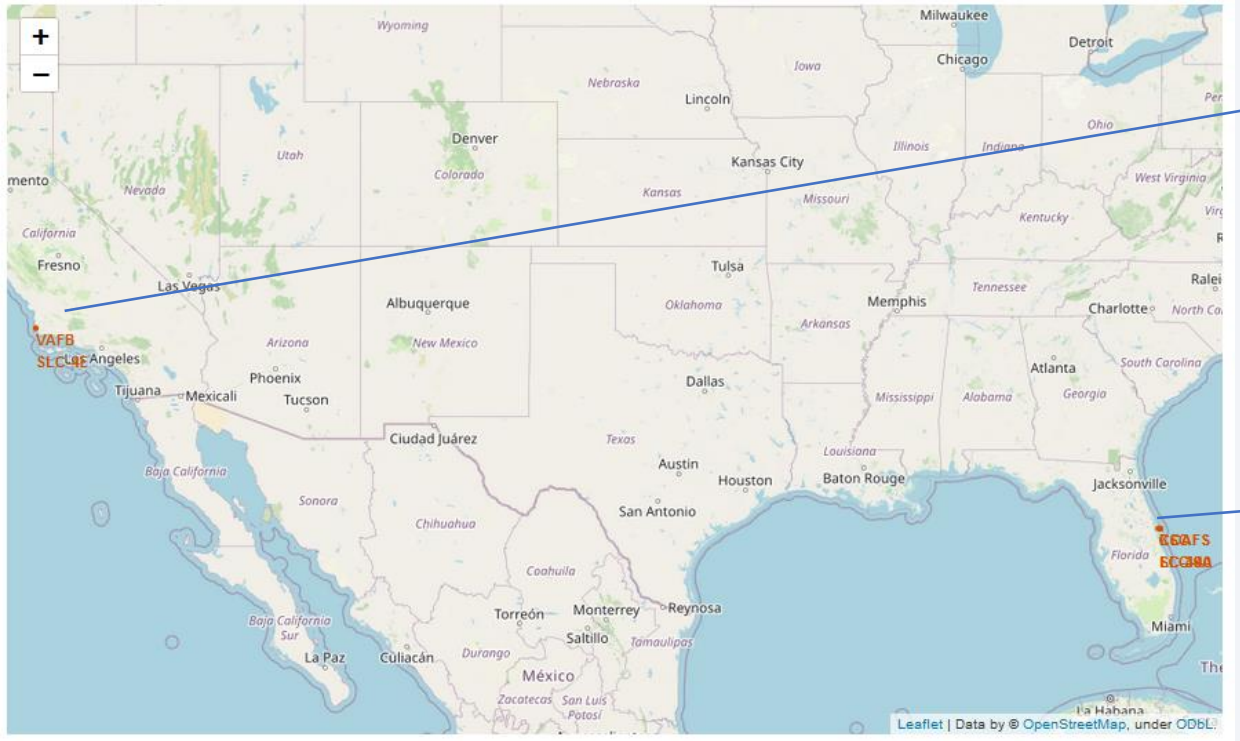
	OUTCOME	TOTAL
0	No attempt	10
1	Failure (drone ship)	5
2	Success (drone ship)	5
3	Controlled (ocean)	3
4	Success (ground pad)	3
5	Failure (parachute)	2
6	Uncontrolled (ocean)	2
7	Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

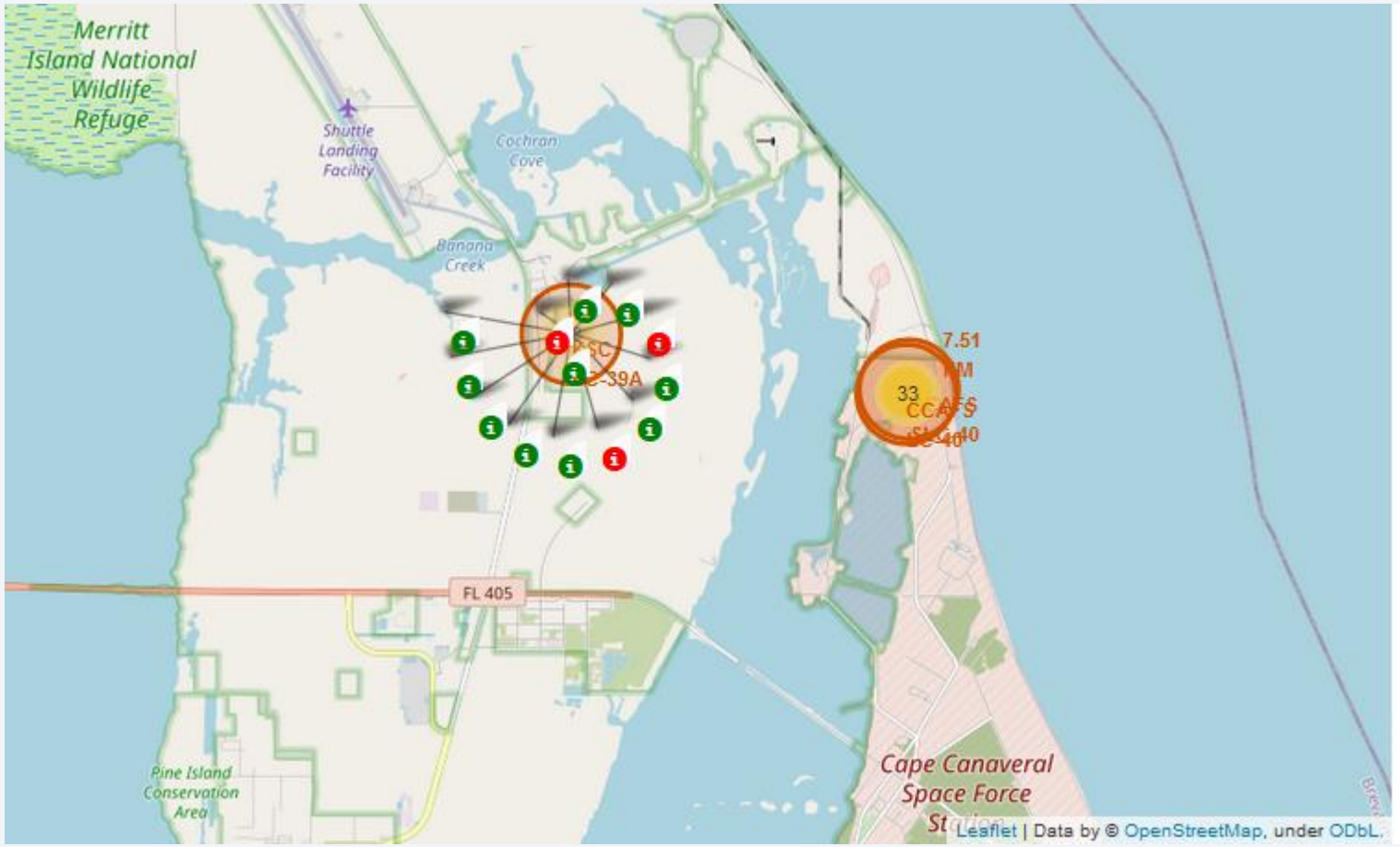
Launch sites location

We created a map visualization for all launch sites location.



Success or failed launches for each site

We created markers for all launch records. If a launch was successful, we use a green marker and if a launch was failed, we use a red marker .

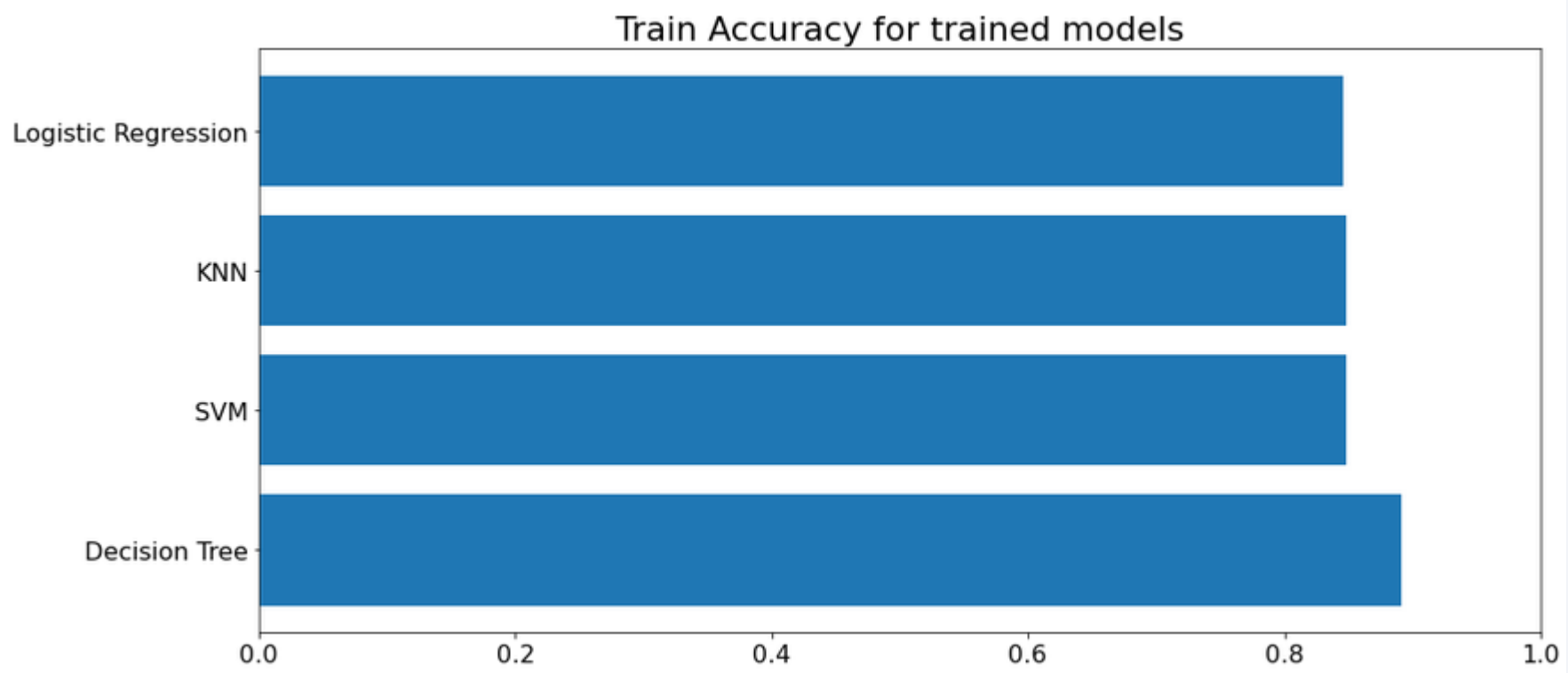


Section 5

Predictive Analysis (Classification)

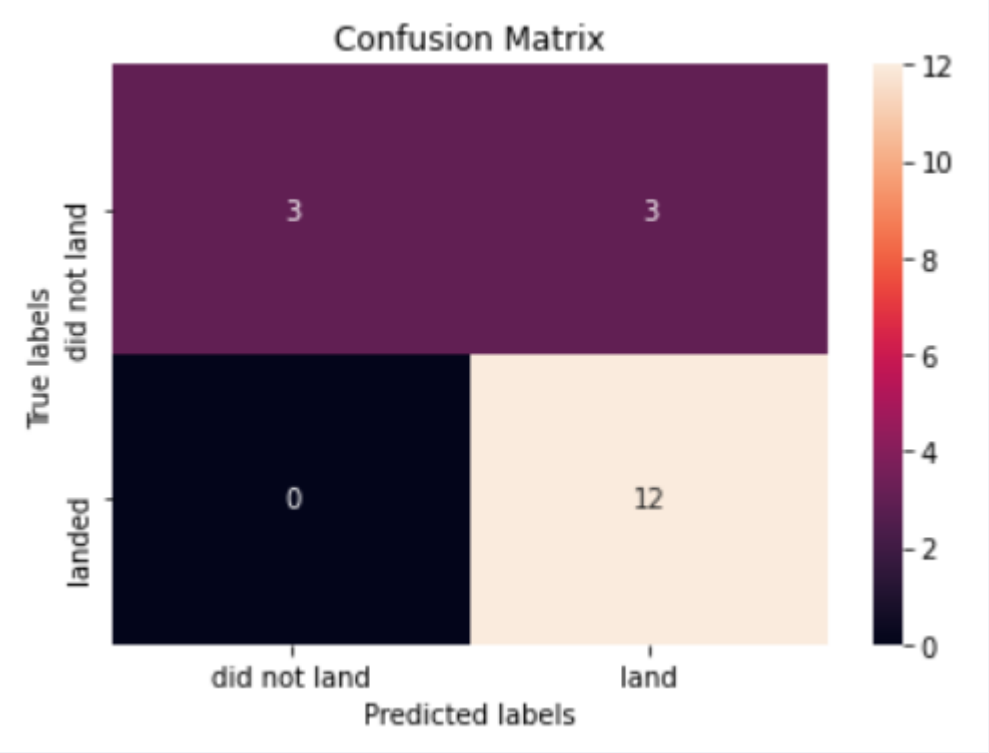
Classification Accuracy

We developed classification models based on Decision Tree, Logistic Regression, Support Vector Machine and K-Nearest Neighbor algorithms.



Confusion Matrix

Decision Tree performed the best results on test data.



Conclusions

- Decision Tree model performed accuracy ~84% on test data;
- For fail outcome the model performed precision = 1, recall = 0.5 and f1-score = 0.67;
- For success outcome the model performed precision = 0.8, recall = 1, f1-score = 0.89;
- The model can predict the landing outcome with a good accuracy.

Appendix

GitHub URLs:

- https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/1-1-spacex-data-collection-api.ipynb
- https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/1-2-webscraping.ipynb
- https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/1-3-data-wrangling.ipynb
- https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/2-1-eda-sql.ipynb
- https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/2-2-eda-dataviz.ipynb
- https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/3-1-launch-site-location.ipynb
- https://github.com/geo-rod/First-stage-falcon9-landing-outcome-prediction/blob/main/4-1-spacex-machine-learning-prediction.ipynb

Thank you!