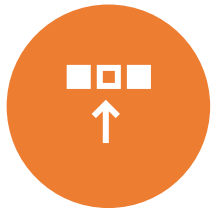
Winning Space Race with Data Science

Jeyson Barrera
May 25, 2022

Outline



EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

APPENDIX

Executive Summary

Data collection

Using an API.

Using website scrapping methodology.

Exploratory Data Analysis (EDA).

EDA with SQL.

EDA with data visualization.

Interactive data visualization using Folium.

Interactive dashboard using Ploty dash.

Predictive analysis.

Introduction

Project background and context

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. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

Problems you want to find answers

What factors determine if the rocket will land successfully?

The interaction amongst various features that determine the success rate of a successful landing

What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

- Data collection methodology.
  - SPACEX API : (API)
  - Web Scraping (Wikipedia)
- Perform data wrangling.
  - Cleaning data and change types if necessary.
- 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

- SPACE X API:
  - Use HTTP request (requests) from request python library to the following consistent endpoint:

    https://api.spacexdata.com/v4/launches/past

    A response code of 200 massive data is retrieved but only will need the following in order to achieve the goal of determinate if as launch will land or not.

    - Booster version, payload mass, longitude and latitude of the launches, orbit, launch site, outcome, number of flight, grind fins, reused, legs, landing pad, block, reused count and serial.

  Before extract, data need to be decoded and presented in a readable format, for this project data is decoded using .json() and turn into a pandas dataframe using .json_normalize().

Data Collection – SpaceX API

1.

```
# 1. HTTP Response from API
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
# 2. Convert response in to json file and
#    pandas dataframe
json_data = response.json()
df_data = pd.json_normalize(json_data)
# 3. Methods to filter data
getBoosterVersion(df_data)
getPayloadData(df_data)
getCoreData(df_data)
getBoosterVersion(df_data)
```

2.

```
# 4. Create dataset in dictionary format
launch_dict = {'FlightNumber': list(df_data['flight_number']),
'Date': list(df_data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
launch_df = pd.DataFrame.from_dict(launch_dict)
```

3.

```
# 5. Filter dataframe and conver it
#    in .csv file
data_falcon9 = launch_df[launch_df['BoosterVersion']!='Falcon 1']
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

GitHub Link - Notebook

Data Collection – Web Scraping

```
# 1. Use requests.get()
html_data = requests.get(static_url).text
# 2. Create Beautiful object
soup = BeautifulSoup(html_data, 'lxml')
# 3. Explore Tables and select the number 2
#    which make reference to the past launches
html_tables = soup.find_all('table')[2]
# 4. Select columns of table
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if(name != None and len(name) > 0):
        column_names.append(name)
```

1.

```
# 5. Create dictionary where data will
#    be appended
del launch_dict['Date and time ( )']

launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

2.

```
# 6.Extract data from table and append it
#   to dictionary.
#########
    # For full detail please see notebook
#########
# 7. Convert data in dataframe and export
#    it to csv file.
df=pd.DataFrame.from_dict(launch_dict)
df_to_csv('spacex_web_scraped.csv', index=False)
```

3.

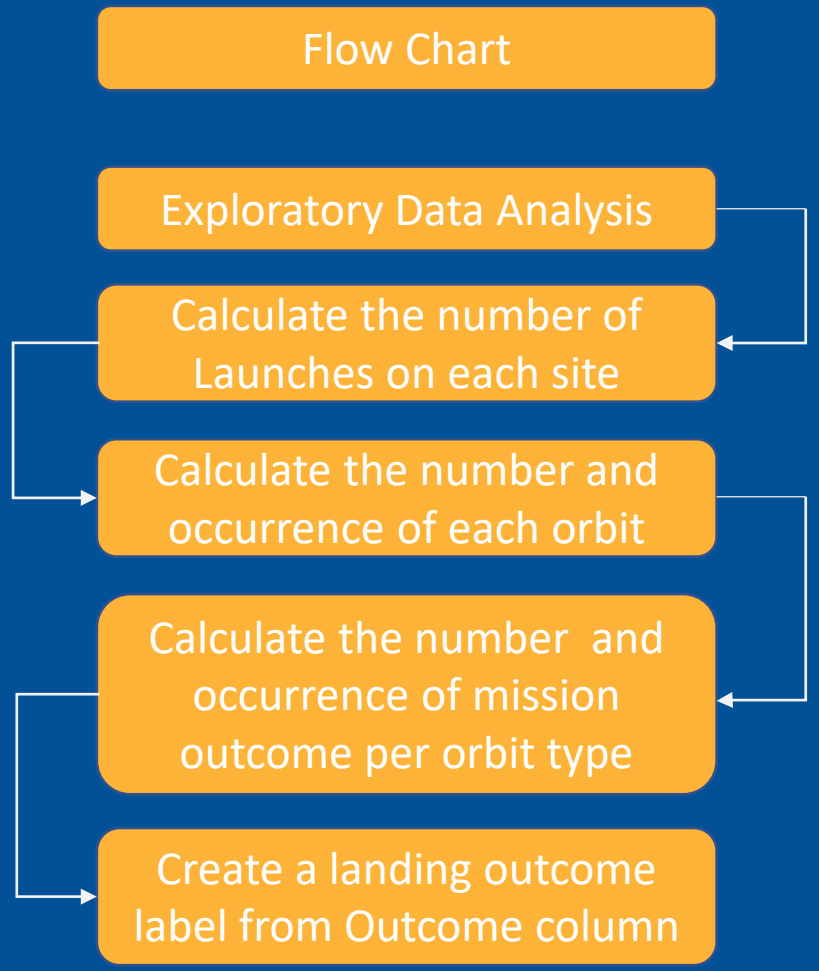
GitHub Link - Notebook

Data Wrangling

The information shown in the data set, allows us to classify a landing as successful or failed landing, as well the place where the landing takes place.

- Successful or failed landing. (True-False landed)
- Place of landing. (Ground pad-drone ship)

With the first scenario is possible convert them into training labels 1 and 0 which means successful and failed, respectively.

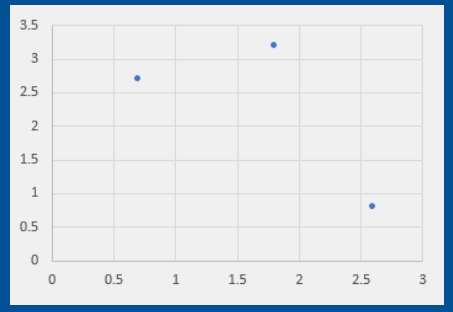


GitHub Link - Notebook

EDA with Data Visualization

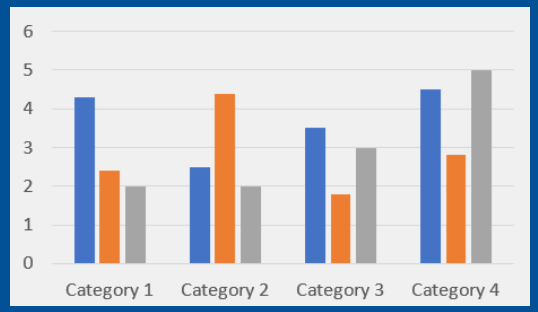
Scatter Plot

- This type of chart is used to determine the relationship between 2 variables (correlation) and how it is affected each other.
- Flight number Vs. Payload Mass
- Flight number Vs. Launch Site
- Payload Vs. Launch Site
- Flight number Vs. Orbit type
- Payload Vs. Orbit type



Bar Plot

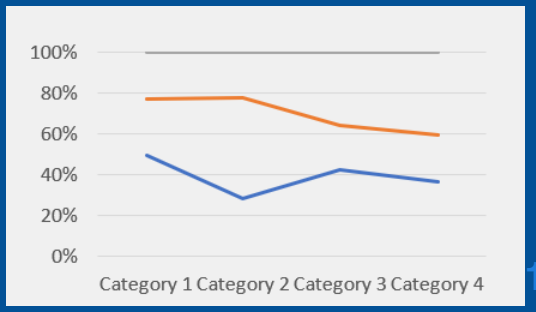
- This type of chart is used to compare two or more data sets and see how they change over time representing categorical data.
- Orbit Vs mean



Line graph.

- This type of chart allows to see trends in data sets over time.
- Date Vs. Success Rate

GitHub Link - Notebook



EDA with SQL

Exploring data querying a Db2 database

- Display the names of the unique launch sites in the space mission.
- Display 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.
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which 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.
- List the names of the booster_versions which have carried the maximum payload mass.
- List the failed landing_outcomes in drone ship, their booster version, and launch site names for in year 2015.
- 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.

GitHub Link - Notebook

Build an Interactive Map with Folium

Folium is a great tool that allowed to visualize the site of each launch giving latitude and longitude. These locations are represented with circle markers and pop ups when a launch site is selected.

Also, for future analysis its necessary show in the interactive map the failure and success of launches with green and red markers.

In addition to this distances between launches and proximity points was calculated to determine if there was any critical point in the case of any failed launch.

After plot distances lines and the proximities following questions were easy to answer:

- Are launch sites in close proximity to railways?
  - No.
- Are launch sites in close proximity to highways?
  - No.
- Are launch sites in close proximity to coastline?
  - Yes.
- Do launch sites keep certain distance away from cities?
  - Yes.

GitHub Link - Notebook

Build a Dashboard with Plotly Dash

For this specific project, Plotly allows to display information in 4 easy steps:

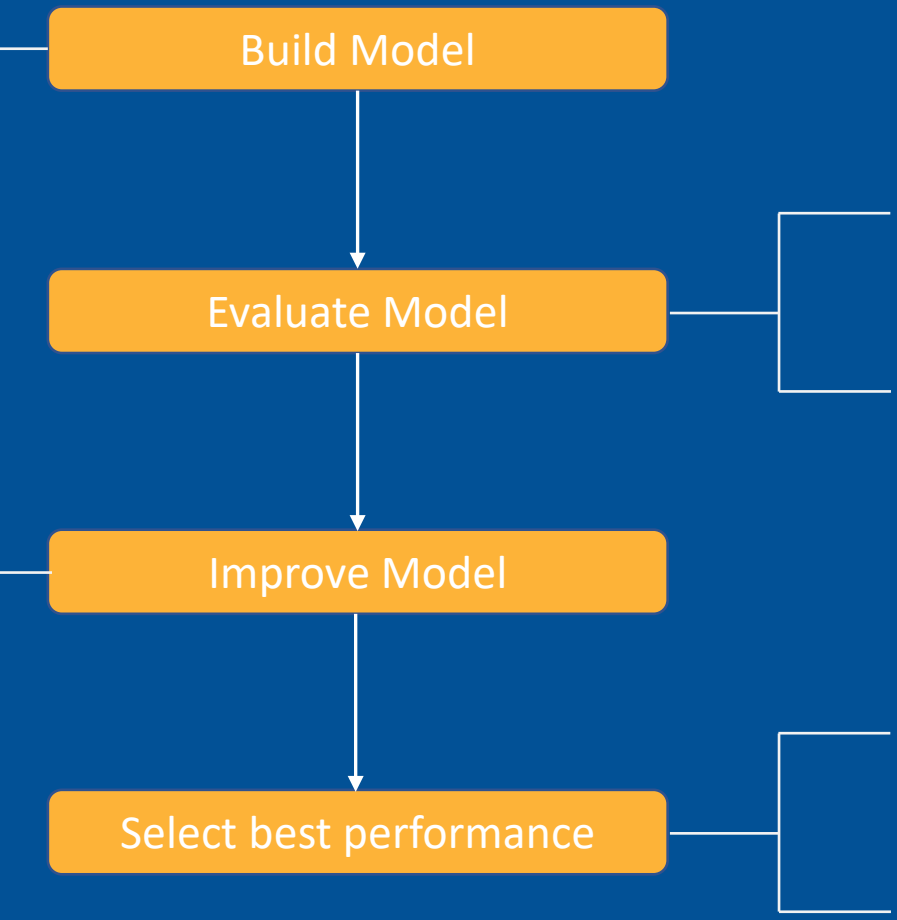
- A drop-down component to select launch site.
- Callback function to render a pie chart.
- A range slider to select a payload.
- Callback function to render success-failed payload in a scatter chart.

Having different launch sites, was important to know what of these sites had a successful payload. For this scenario, its easy to classify data (0, 1) for failed and successful, respectively.

To see the relationship between the outcome and payload mass for different booster version is necessary use a scatter graph.

Predictive Analysis (Classification)

- Load datasets.
- Clean data.
- Split data (training-test).



- Check for accuracy.
- Generate confusion matrix.

- Algorithm Tuning.

- Best performing model.

GitHub Link - Notebook

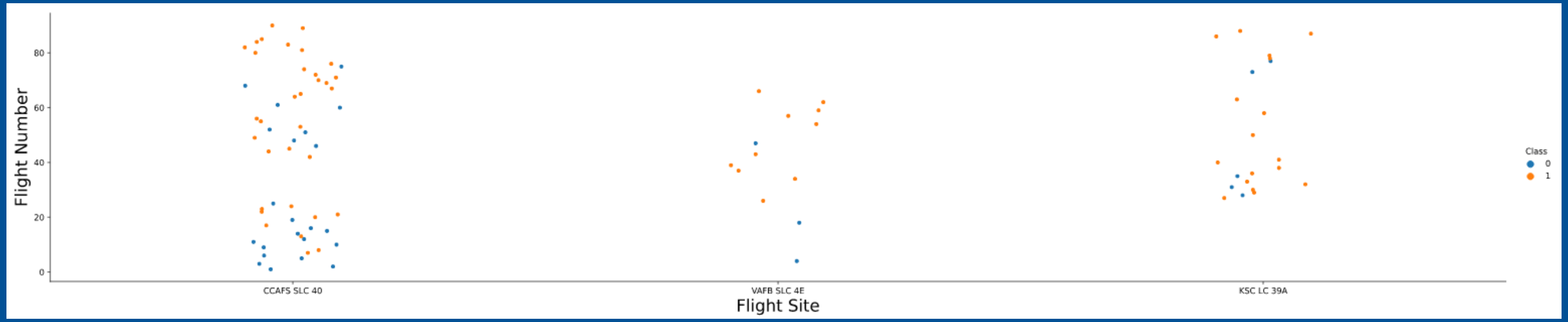
Results

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

Section 2

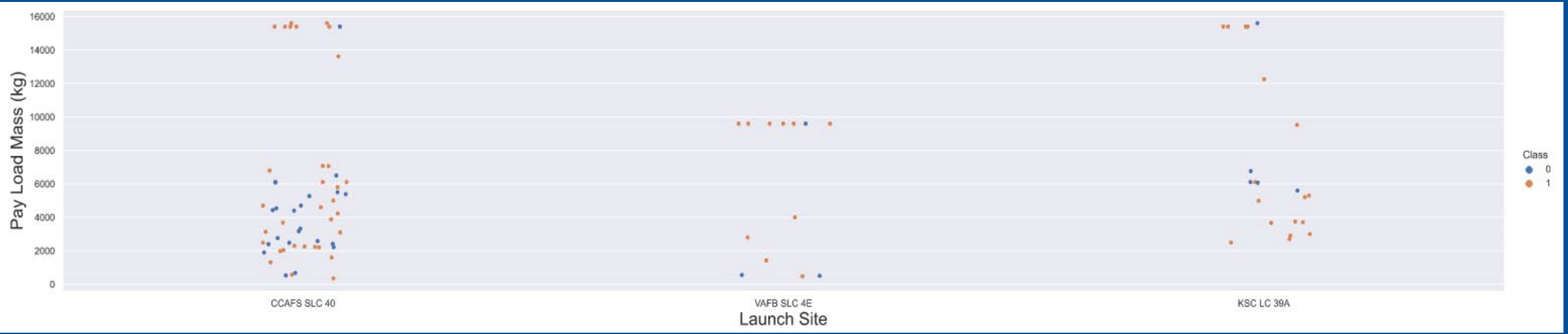
Insights drawn from EDA

Flight Number vs. Launch Site



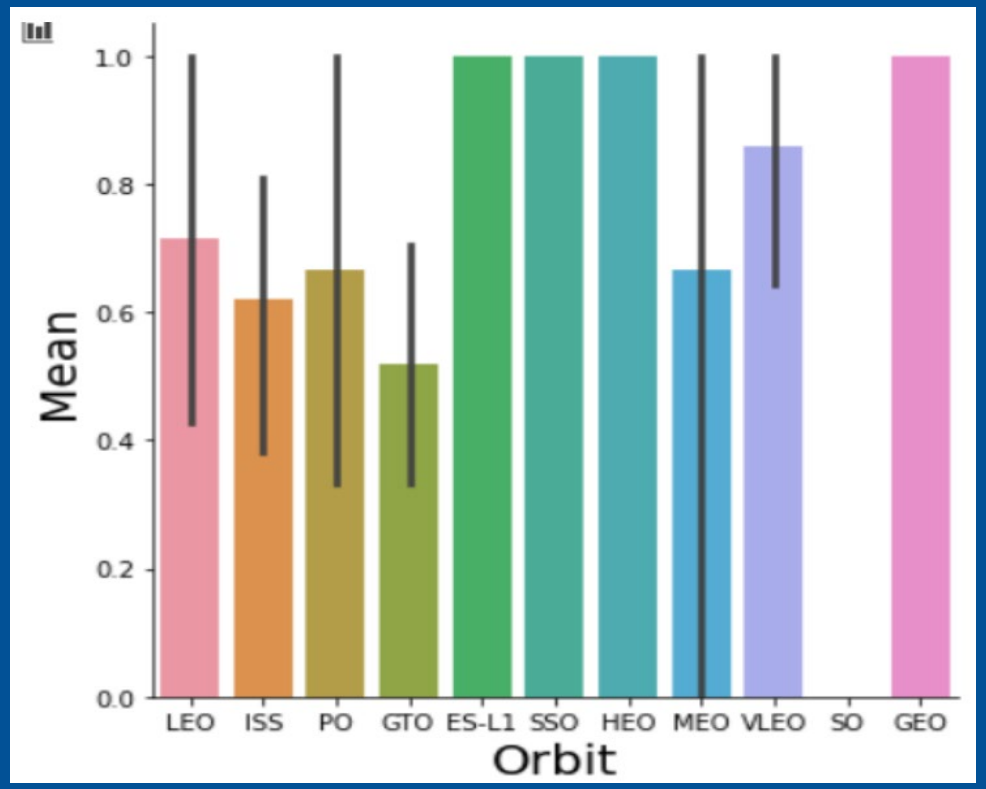
According to the graph CCAF5 SLC 40 launch site
has great successful rate of launches.

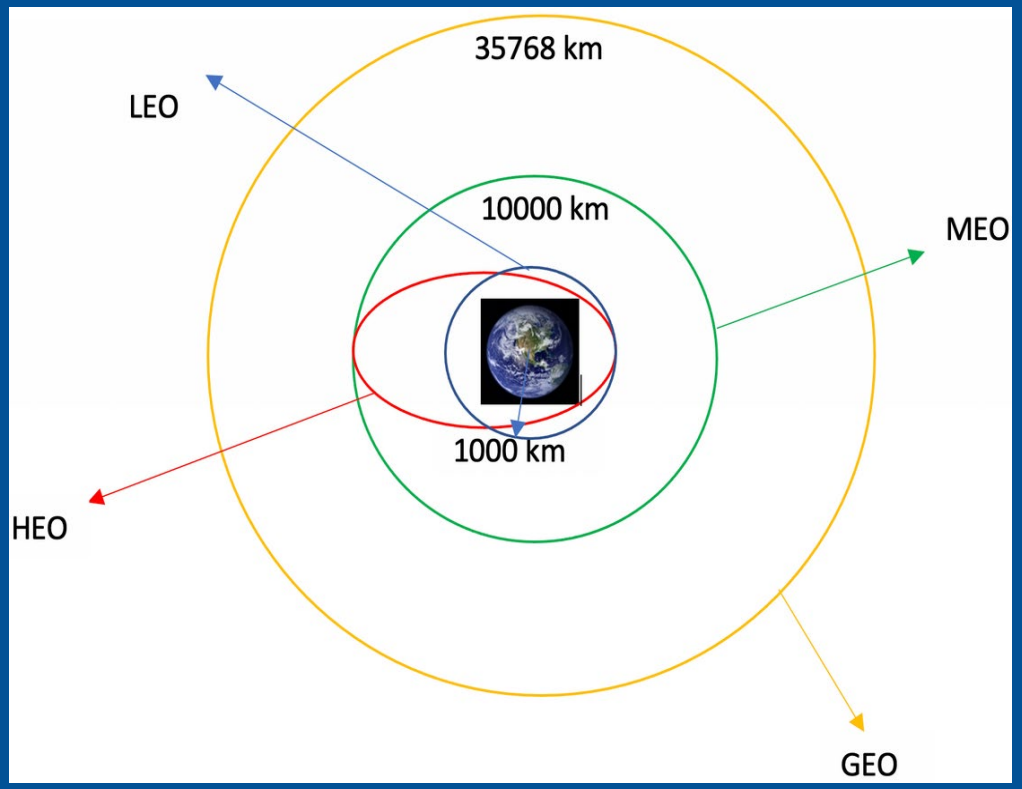
Payload vs. Launch Site



Its possible to infer that the higher the pay load mass at CCAFS SLC 40 the higher rate of success, but the datasets is showing a lot disperse datapoints. That's why is good try to analyze this affirmation more in depth.

Success Rate vs. Orbit Type

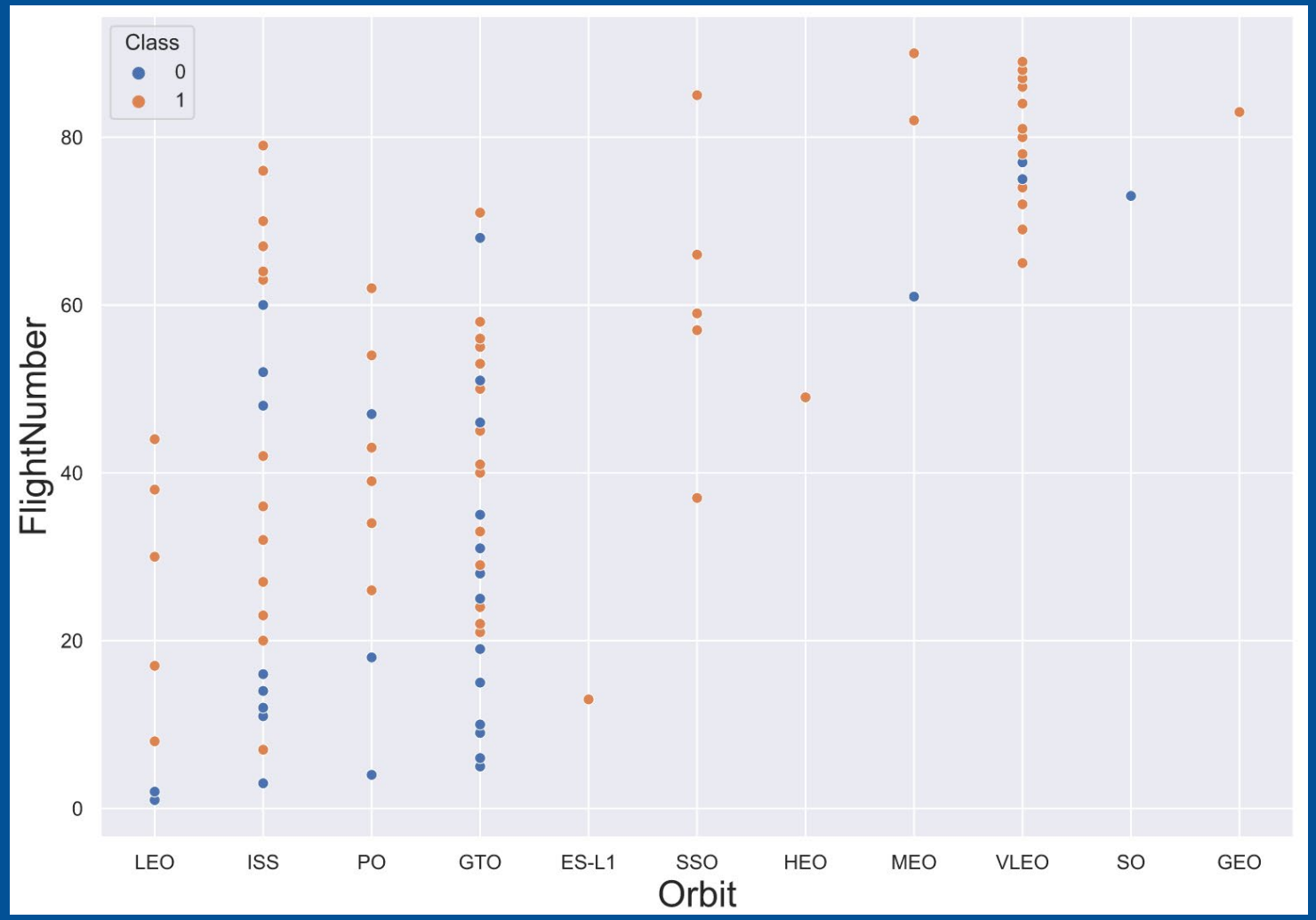




Is possible affirm that orbits ES-L1- SSO - HEO and GEO, are the orbits where most off the successful launches occur.

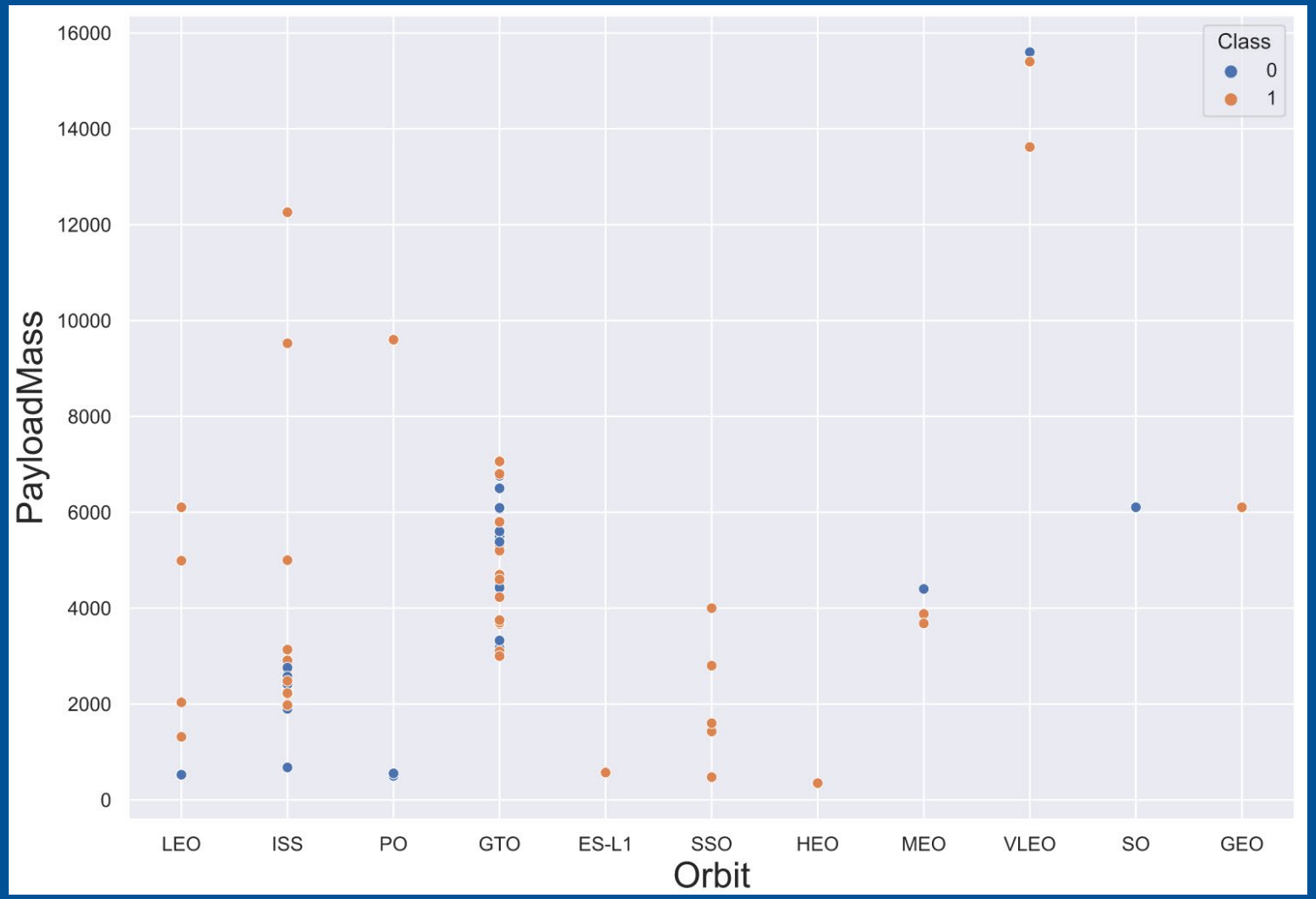
Flight Number vs. Orbit Type

According to the graph, VLEO orbit seems to have a relationship with the flight number.



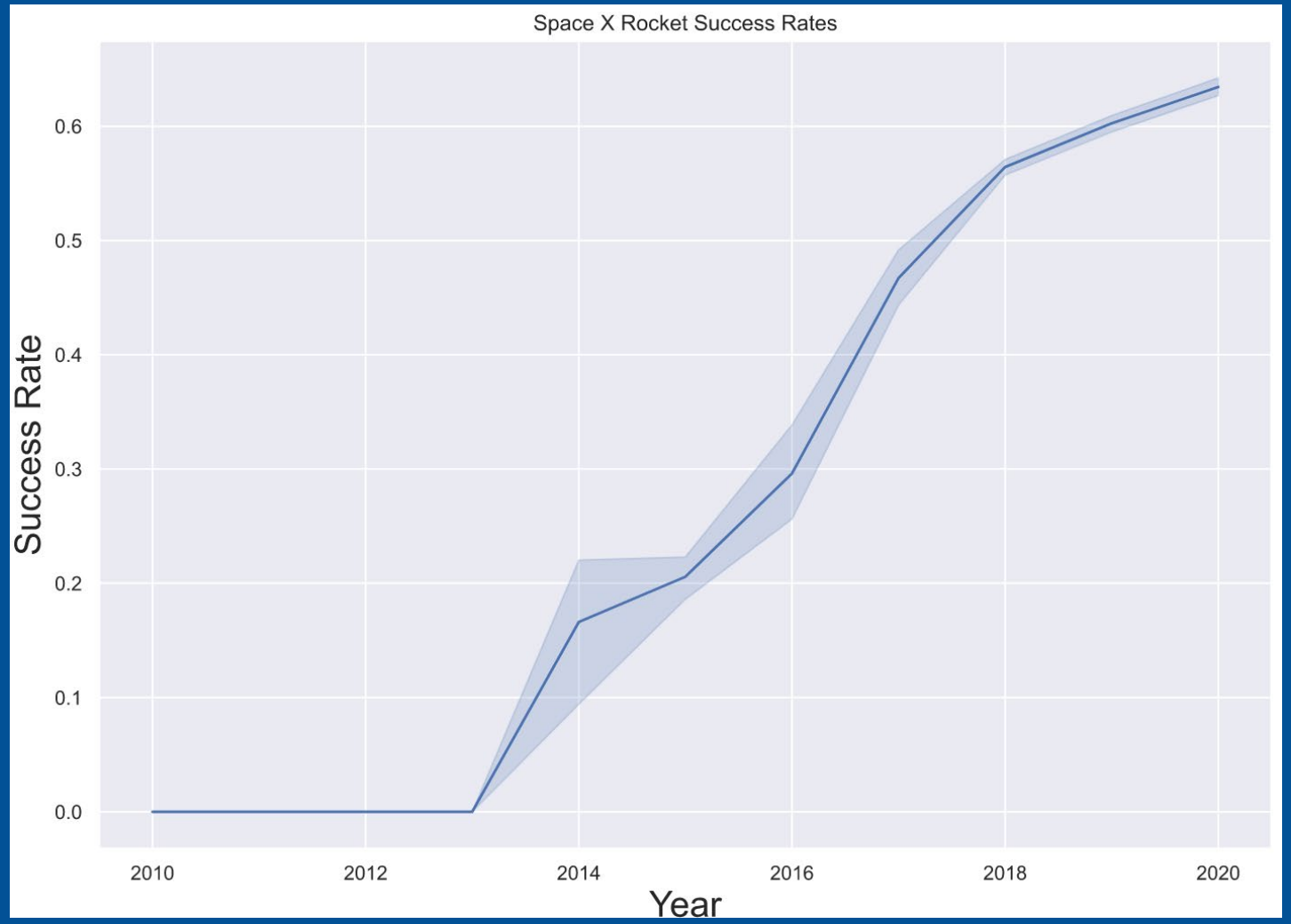
Payload vs. Orbit Type

Is possible to affirm that payload mass has considerable rate of failed launches on orbit GTO.



Launch Success Yearly Trend

With this line graph is clear that launches have been improving constantly since 2013.



All Launch Site Names

Using the statement DISTINCT in the query to the database the result guarantee that unique values are Retrieved.

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

```
 * ibm_db_sa://

Done.
```

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

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

```
 * ibm_db_sa://
Done.
```

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

Using the statement LIMIT in the query to the database the result guarantee only 5 entries will be shown.

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://
Done.
    1
45596
```

Using the statement SUM in the query to the database the result will be a total mass in kilograms of all payloads

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://
Done.
```

1
2928.400000

Using the statement AVG in the query to the database the result will be an average of mass in kilograms of all payloads.

First Successful Ground Landing Date

```
%sql select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://
Done.
```

1
2015-12-22

Using the statement MIN in the query to the database the result will be the first date of a successful landing.

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_ > 4000 and PayloadMassKG < 6000'
```

```
 * ibm_db_sa://
Done.
```

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

Using the ranges between 4000 and 6000 in the payload mass a list of booster version will be displayed.

Total Number of Successful and Failure Mission Outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where \
MISSION_OUTCOME = 'Success' or \
MISSION_OUTCOME = 'Failure (in flight)'
```

```
 * ibm_db_sa://
Done.
  1
100
```

A combination and filter of two keywords ‘Success’ and ‘Failure’ in the mission outcome column a total of 100 mission happened.

Boosters Carried Maximum Payload

```
%sql select BOOSTER_VERSION from SPACEXTBL where \
PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) \
                     from SPACEXTBL)
```

```
 * ibm_db_sa://
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

Using a subquery and finding the maximum (MAX) payload mass 12 names has the maximum Weight in kilograms

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

```
%sql select * from SPACEXTBL where Landing__Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```

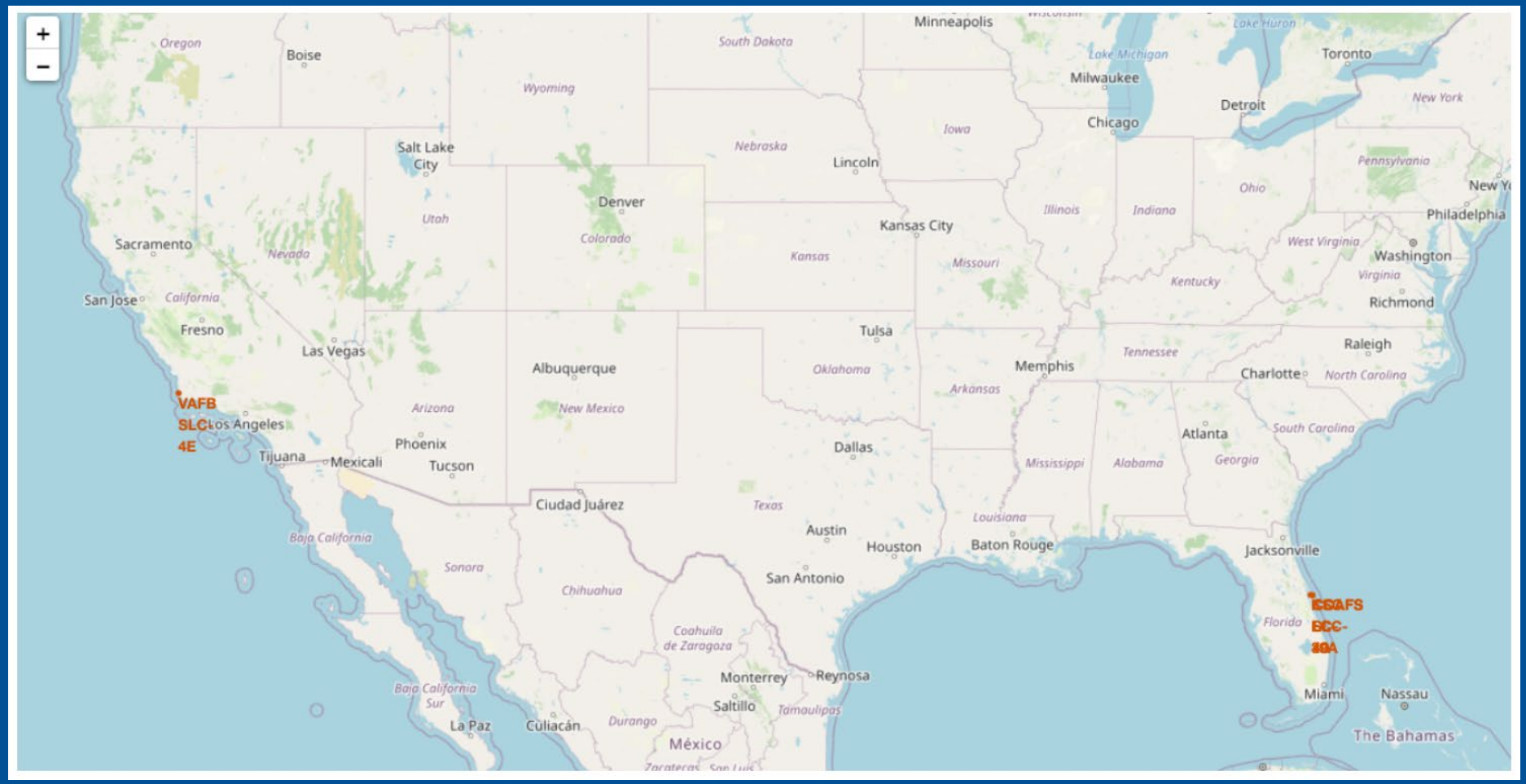
```
 * ibm_db_sa://
Done.
```

DATE	time_utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Slicing the query to the database between June 04, 2010 and Marc 06, 2017 8 flights occurred.

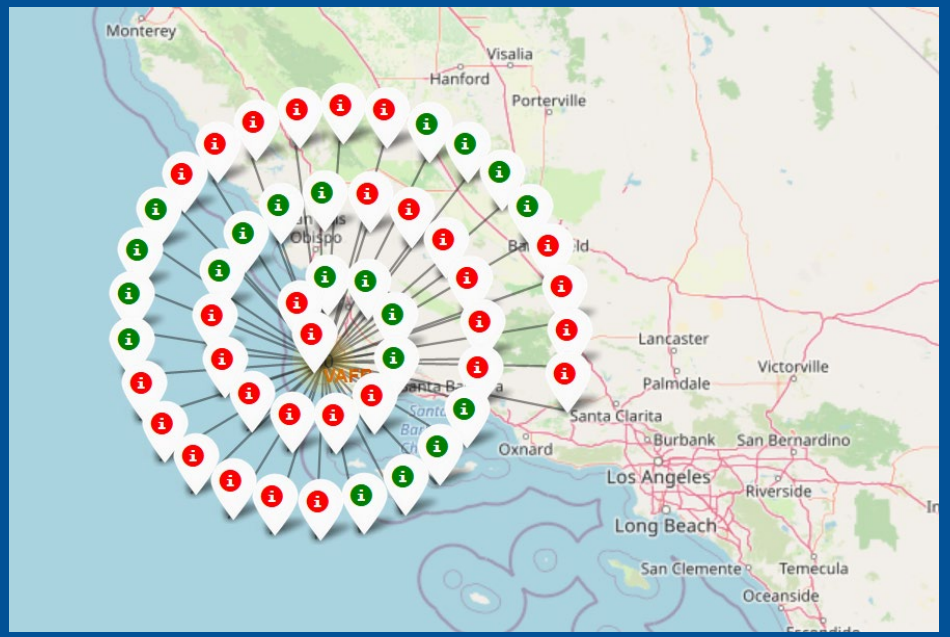

Section 3
Launch Sites
Proximities Analysis

Space X launch sites

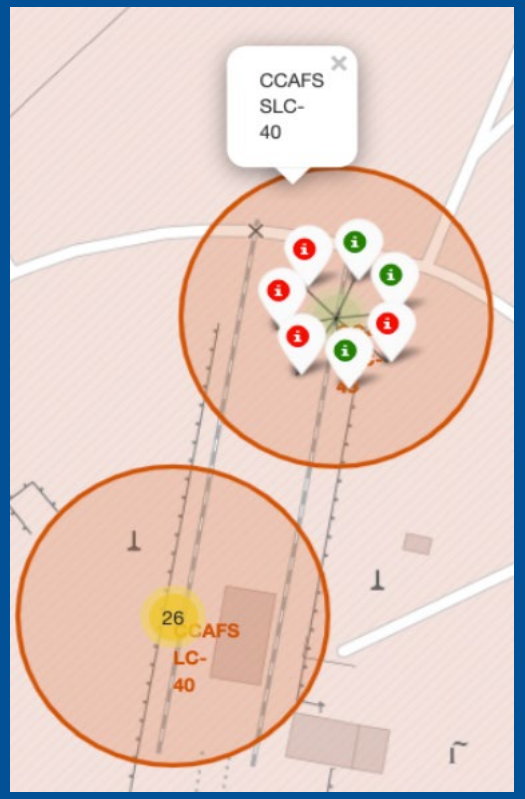


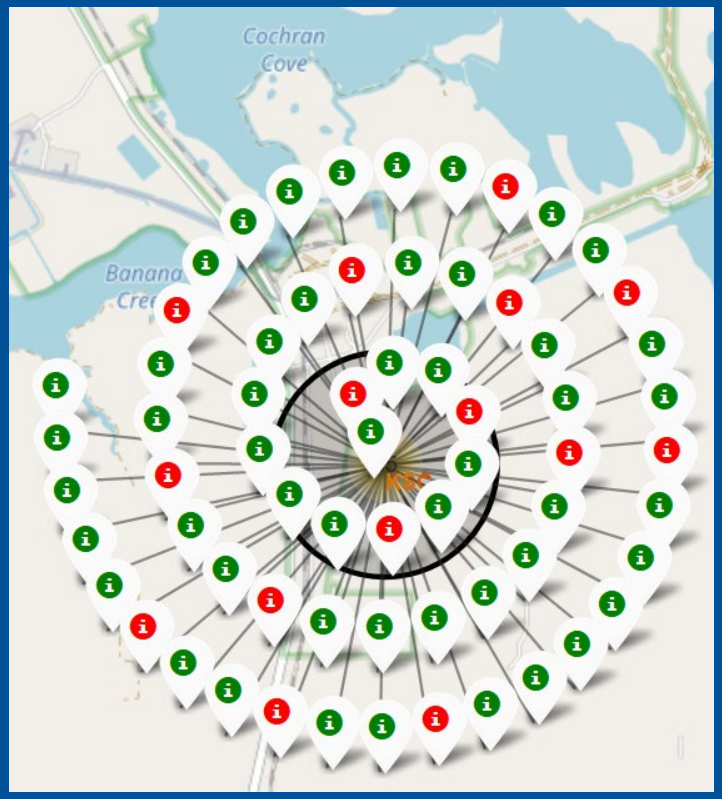
Is right to say that launches are in proximity of the coast for safety reasons in case of any dangerous situation.

Successful – Failure launches



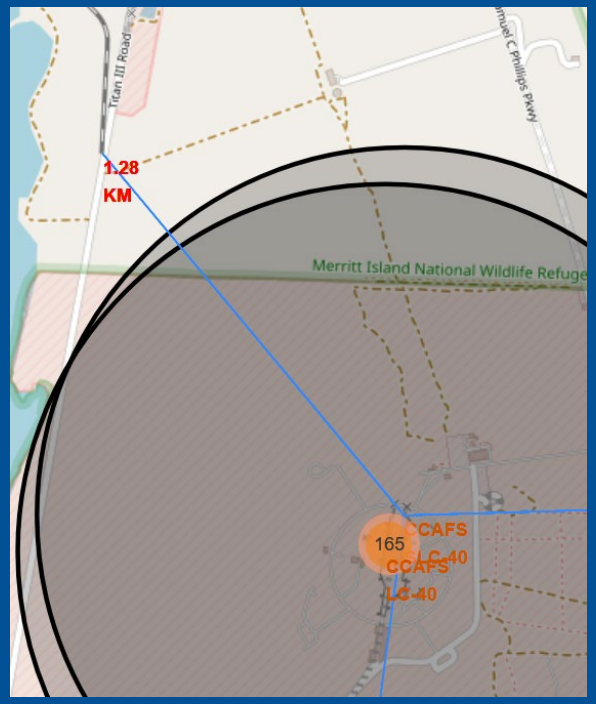
California launch site



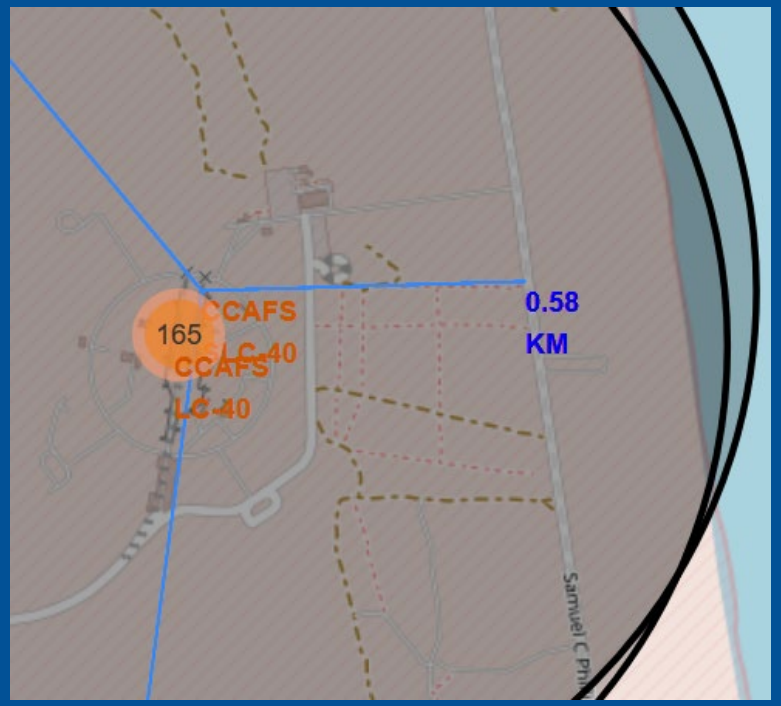


Florida launch sites

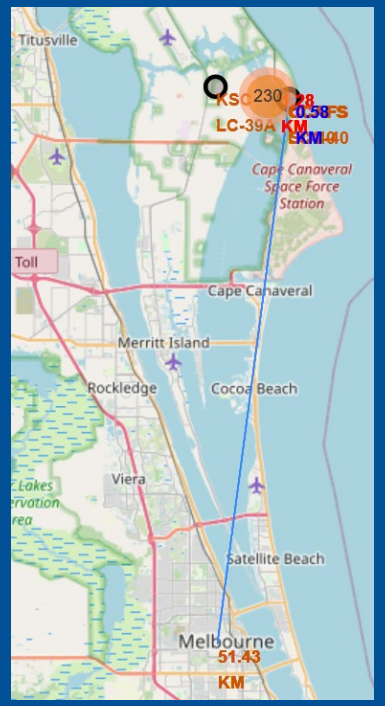
Distance from launch site to critical points



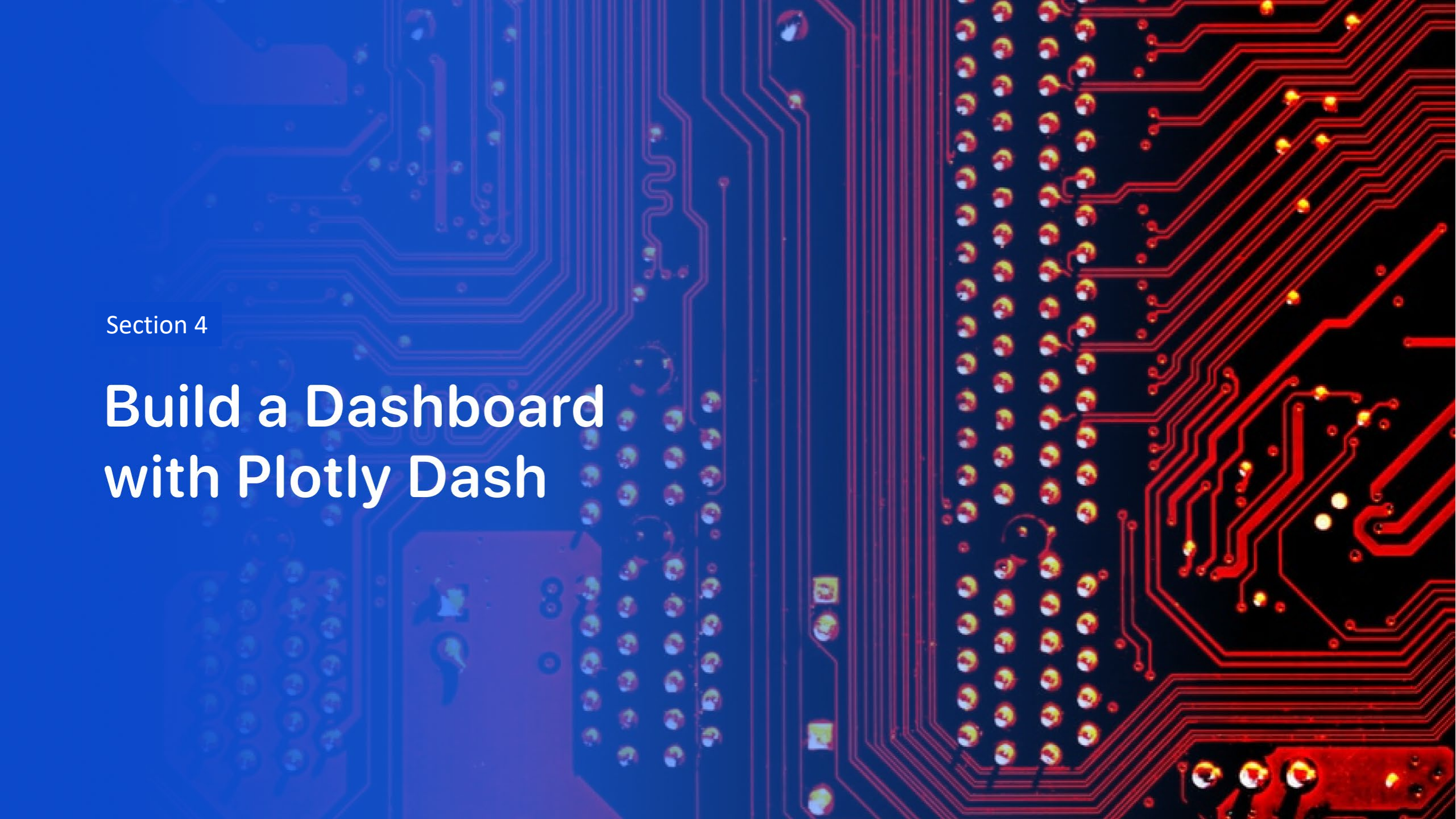
Distance to railways



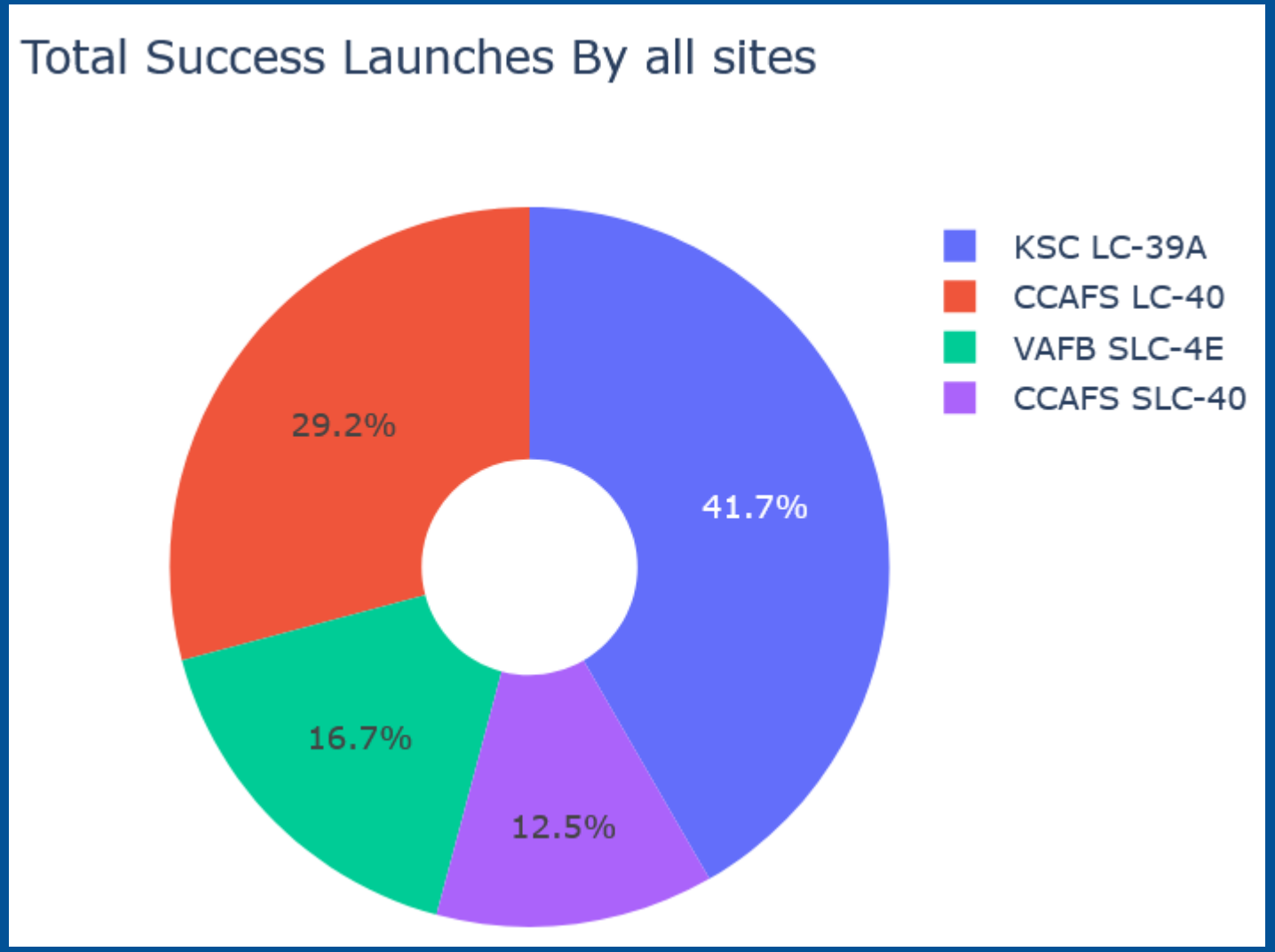
Distance to highway



Distance to nearest city

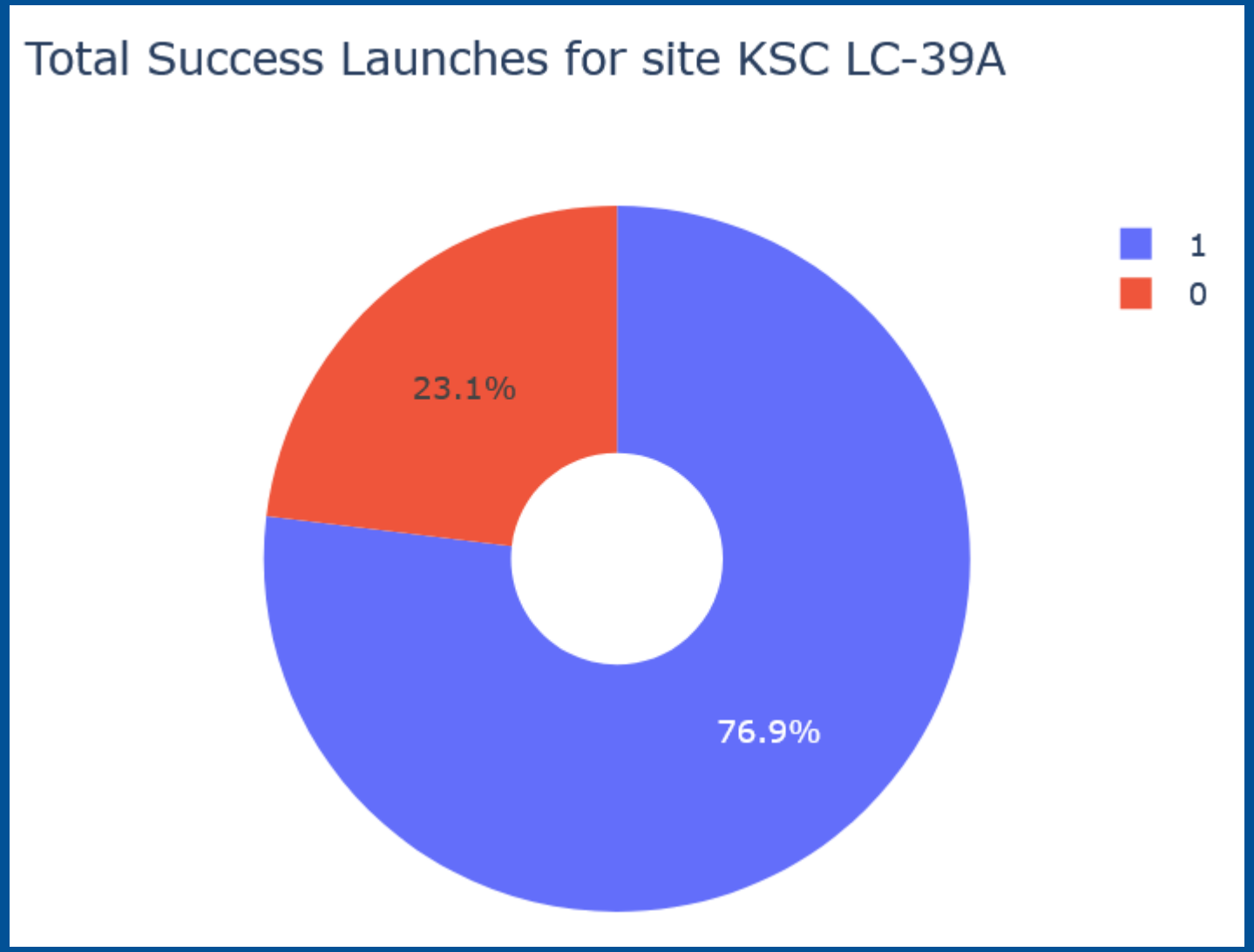

Section 4
Build a Dashboard with Plotly Dash

Success Launches Dashboard – All stations



The pie chart shows that the station KSC LC-39A has the most successful launches with a 41.7% in comparison with CCAFS SLC-40 that has the worst successful rank with only 12.5% of the cases.

Success Launches Dashboard – KSC LC-39A



The pie chart shows that the station KSC LC-39A has 76.9% success rate and only 23.1% failure rate

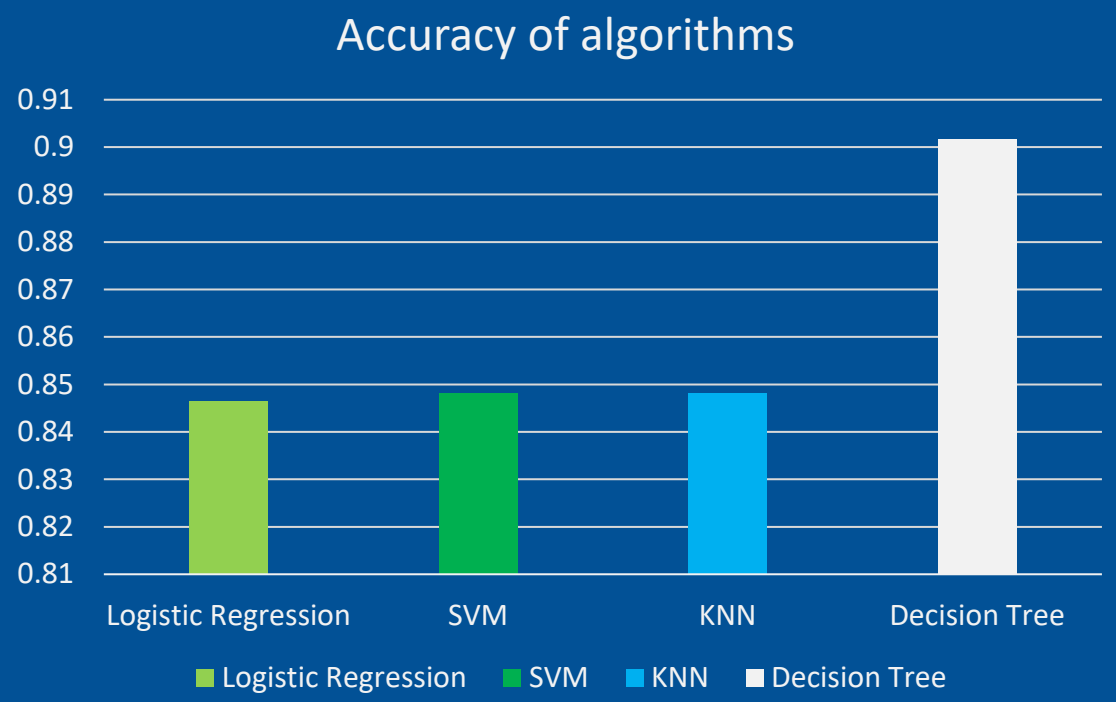
Dashboard – Payload vs Launch outcome

This two-scatter plot confirms that station KSC LC-39A has more successful launches in relation with the payload mass.

Section 5

Predictive Analysis (Classification)

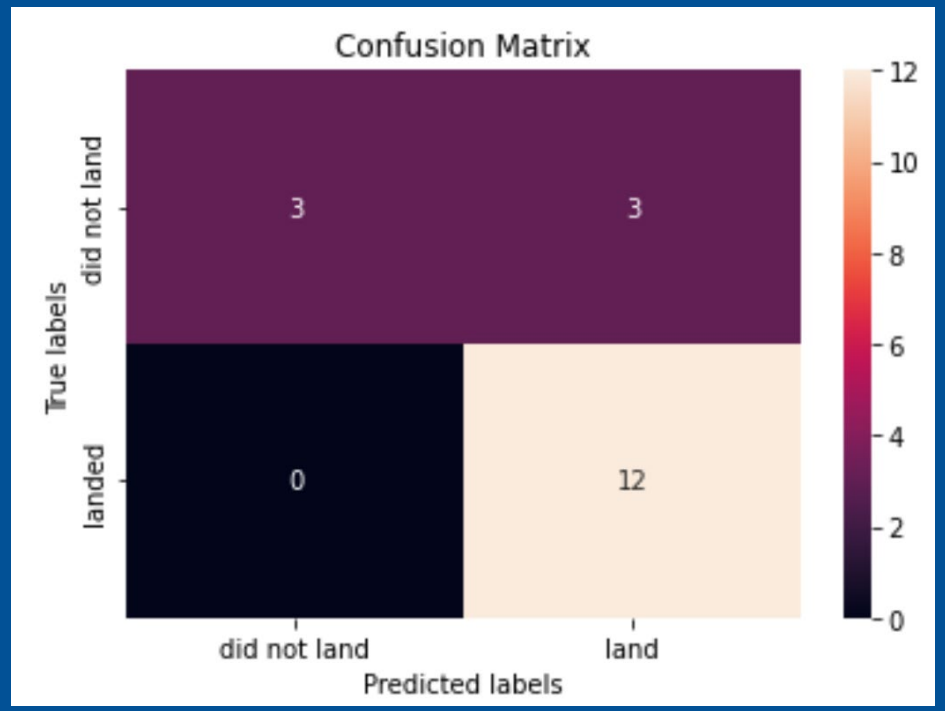
Classification Accuracy

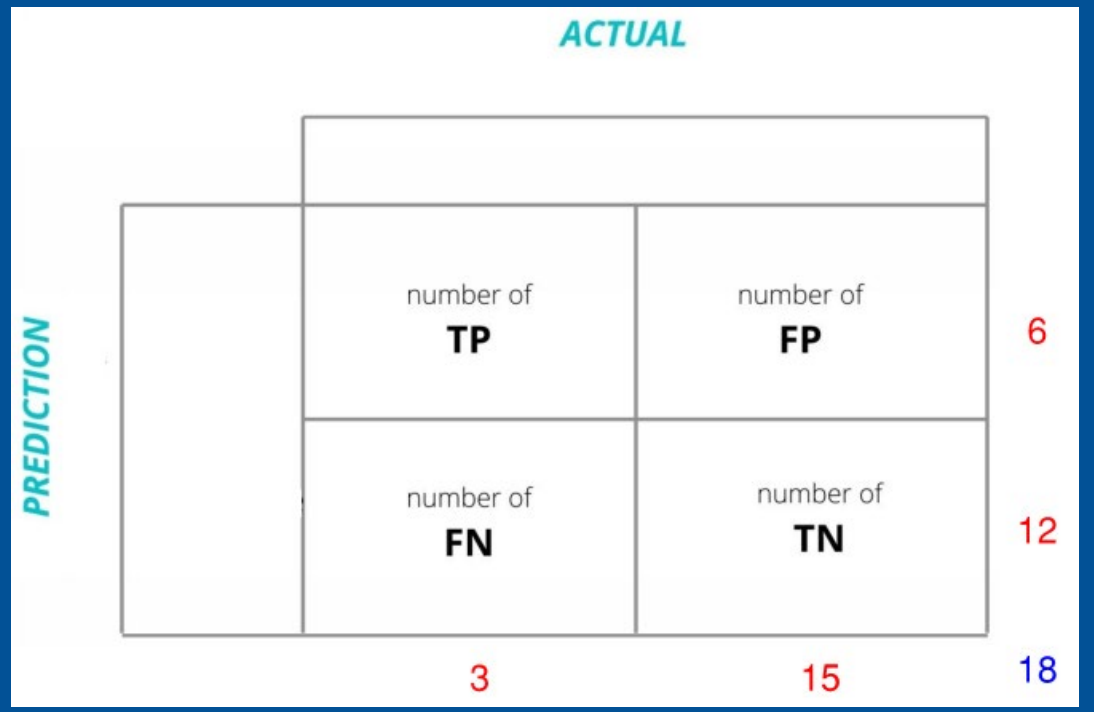


According to the chart logistic regression, SVM, KNN have a similarity with an 84% of accuracy, but decision tree has the best performance with 90% of accuracy.

Confusion Matrix

Confusion Matrix for Decision Tree Algorithm





Accuracy: (TP+TN)/Total = (12+3)/18 = 0.83
Misclassification Rate: (FP+FN)/Total = (3+0)/18 = 0.16
True Positive Rate: TP/Actual Yes = 12/12 = 1
False Positive Rate: FP/Actual No = 3/6 = .5

True Negative Rate: TN/Actual No = 3/6 = .5
Precision: TP/Predicted Yes = 12/15 = .8
Prevalence: Actual Yes/Total = 12/18 = .6

Conclusions

Rocketship launches has been increasing since 2013 steadily. It looks like through time SPACE X is improving making them achieve their goal.

Orbits ES L1, GEO, HEO, SSO have the best success rates of launches.

Decision tree algorithm is the best for the dataset provided.

Increasing payload mass try to lower the rate of the success launches.

Appendix

Most important queries used in EDA – SQL Module

- Display the total payload mass carried by boosters launched by NASA (CRS)
  - select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - select BOOSTER_VERSION from SPACEXTBL where \ PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) \ from SPACEXTBL)

Appendix

```
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
grid_search = GridSearchCV(svm, parameters, cv=10)
svm_cv = grid_search.fit(X_train, Y_train)
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

Vector machine to
create a GridSearchCV
and tune
hyperparameters.

Thank you!