IBM Developer
SKILLS NETWORK

WINNING SPACE RACE WITH DATA SCIENCE

SIXTO MARTINEZ
July 19th, 2022

OUTLINE

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

Executive Summary

The purpose of the project is to predict the outcome of the 1st stage landing for the SpaceX Falcon 9 rocket, which will finally used to determine the cost of the launch by knowing if it could be potentially reused or not.

To be able to predict the successful landing or not of these launches, several machine learning models were developed and trained with public information. The models used for the prediction were: Logistic Regression, SVM, Classification Trees & K-Nearest Neighbor.

Summarizing, the method that proved to be the most accurate using the training data collected was the Decision Tree Classifier with a Test Accuracy of approximately 94.4%. The rest of method used had a Test Accuracy of 83.3% when tested with a test data after being trained with the train data.

It was also noted that there is a certain correlation between the outcome of the launch and other parameters like Launch Site, Payload Mass, Launch Year, among others.

Introduction

Falcon 9 is a reusable, two-stage rocket designed and manufactured by SpaceX for the reliable and safe transport of people and payloads into Orbit and beyond. The reason why SpaceX has been so successful is the low cost of its services due to the reusability of the 1st stage of the rocket launch, which constitutes the most expensive part of the rocket.

Of course, there are launches where it is not possible to reuse the 1st stage of the rocket because the landing of the booster has not been successful.

Therefore, if the successful landing of the 1st stage of the rocket (the booster) can be estimated and its relation to other parameters that can be controlled, there is a high probability that the cost of the launch can be determined.

All these elements are helpful information for the mission of the company as it is trying to compete with SpaceX with the development of a more cost-effective space launch.

METHODOLOGY

Methodology – Executive Summary

DATA COLLECTION

Data was collected using SpaceX's Rest API and Web Scrapping from Wikipedia sites.

DATA WRANGLING

- Accounted only data from Falcon 9 launching data excluding all data related to Falcon 1
- Replaced columns missing values with columns means
- Used One Hot Encoding to transform categorical variables into factors
- Classify landing outcomes as Successful (category 1) and Failed (category 0)

DATA ANALISIS WITH SQL AND VISUALIZATION TOOLS

- Used SQL queries to navigate through the collected data
- Used Scatter Plots and Bar Graphs to evaluate variables dependencies

VISUAL ANALYTICS USING FOLIUM AND PLOTLY DASH

- Analyze launch sites using Folium Maps
- Analyze launch parameters at each launch site using Plotly Web Dash

PREDICTIVE ANALYSIS USING CLASSIFICATION MODELS

- Trained and test classification models to predict landing outcomes with the data collected
- Selected the most accurate model based on prediction's accuracy

Methodology – Data Collection

The data collected and wrangled (refined) to train and test the prediction models was taken from two different sources:

1. SPACEX REST API

The API used was https://api.spacexdata.com/v4/cores to request the data from URL=https://api.spacexdata.com/v4/launches/past.

The response was decoded using *.json()* and turned into a dataframe using *.json_normalize()*

Use API to gain information using launches ID

Parameters used to standardize data:
Rocket = Booster Name
Payload = Payload Mass and orbit that is going to
Launchpad = name of launch site
Cores = landing outcome

```
data_falcon9 = launch_df.loc[launch_df['BoosterVersion']!='Falcon 1']
```

Create dataframe with desired dictionary parameters & filter the dataframe to only include Falcon 9 launches

Replaced column's missing values with the means of each column

```
data_falcon9['PayloadMass'].replace(
        to_replace=np.nan,
        value=PayloadMass_mean,
        inplace=True)
```

Export the data to CSV

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Link to GiHub Notebook:
data-final-project/Data Collection API.ipynb at main · SixtoM/data-final-project (github.com)

Methodology – Data Collection

2. WEB SCRAPPING

Performed an HTTP GET method to request Falcon 9 data from a Wikipedia website

```
static_url_wiki = requests.get(static_url)
```

Created a BeatifulSoup object from the HTML response

```
soup = BeautifulSoup(static_url_wiki.text, "html.parser")
```

Find all tables from the wiki page and extract column names

Create dictionary with same parameters as data taken with SPACEX API and fill the dictionary with data from extracted tables

Create a dataframe from the recently filled dictionary

Export data to a CSV file

```
df.to_csv('spacex.web_scraped.csv', index=False)
```

Link to GiHub Notebook:
data-final-project/Data Collection with Web Scrapping (1).ipynb at main · SixtoM/data-final-project (github.com)

Methodology – Data Wrangling

The purpose of this stage was to treat the data in the best way possible for the training and later testing of the models as well of the results analysis.

There were several outcome categories called “Training Labels” that needed to be converted into simpler parameters for easier handling. Finally, the outcome categories were classified as only two (2): Successful (category 1) and Failed (category 0)

WRANGLING PROCESS

STEP 1

DATA LOAD

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")
```

STEP 2

Identify missing values and column types

```
df.isnull().sum()/df.count()*100
```

```
df.types
```

Link to GiHub Notebook:
data-final-project/Data Wrangling.ipynb at main · SixtoM/data-final-project (github.com)

Methodology – Data Wrangling

WRANGLING PROCESS

STEP 3

OBTAIN TRAINING LABELS DATA & IDENTIFY DESIRED PARAMETERS

```
landing_outcomes = df['Outcome'].value_counts()
```

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

STEP 4

CREATE DESIRED LANDING OUTCOME LABELS

```
landing_class=[]
for key, value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

STEP 5

EXPORT THE DATA TO A CSV FILE

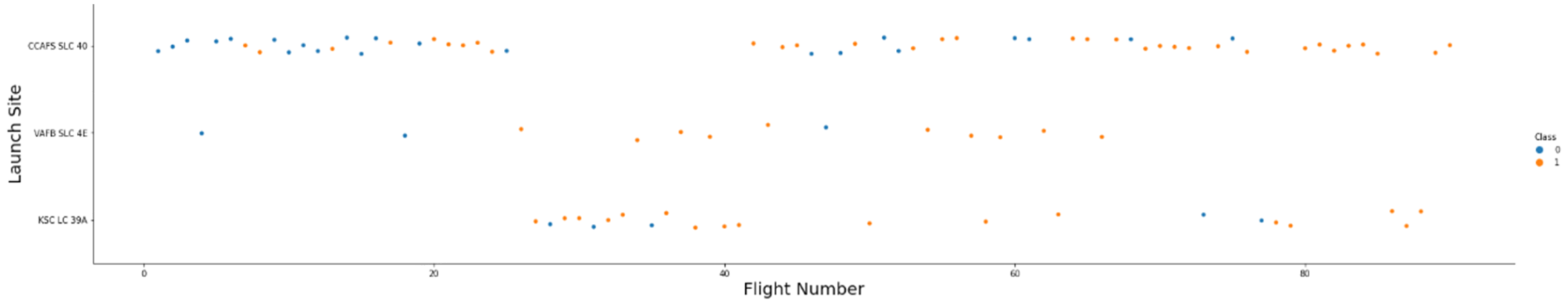
```
df.to_csv("datatset_part_2.csv", index = False)
```

Link to GiHub Notebook:
data-final-project/Data Wrangling.ipynb at main · SixtoM/data-final-project (github.com)

EXPLORATORY DATA ANALYSIS WITH VISUALIZATION TOOLS

Trying to find the correlation between multiple factors and parameters that could impact the outcome of the landing of the first stage of the rocket, several graphs were plotted to understand the interactions between these parameters.
The following graphs were plotted for analysis using the data previously collected and refined:

1) FLIGHT NUMBER VS LAUNCH SITE:

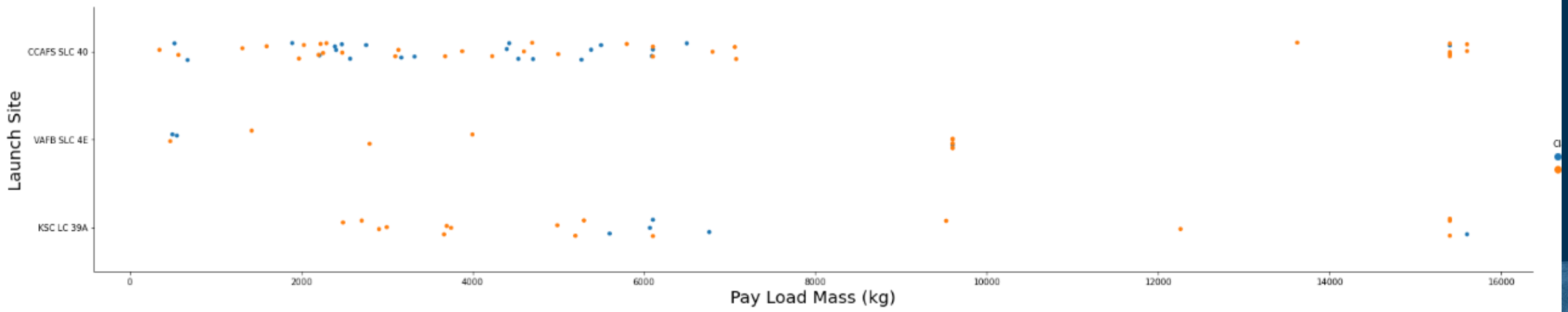


This graphs does not show a significant relationship between CCAFS SLC40 launch site and number of flights that affects the outcome of the landing. However, it does show tendency for of successful landings when launched from VAFB SLC4E & KSC LC39A as the number of flights increases.

Link to GiHub Notebook:
data-final-project/EDA with Visualization Lab.ipynb at main · SixtoM/data-final-project (github.com))

EXPLORATORY DATA ANALYSIS WITH VISUALIZATION TOOLS

2) PAY LOAD MASS VS LAUNCH SITE:



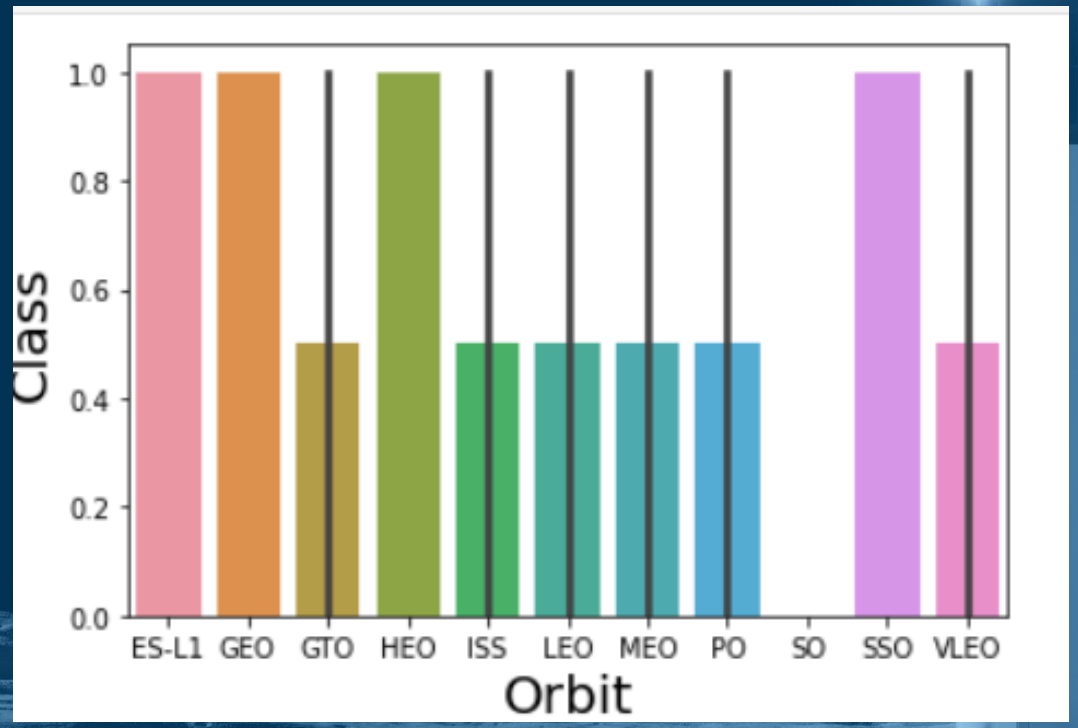
Trying to anticipate a relationship between launch sites and pay load masses with the outcome of the launches, the following findings were observed:

- No launches were made from VAFB SLC4E where the payload mass was over a 10000 Kg.
- Launches with Pay Load Mass > 10,000 Kg were made from CCAFS SLC40 & KSC LC39A with mostly successful landings.
- Landing of the rockets when the Pay Load Mass < 8,000 Kg did not present a tendency to be either successful or failed.

Link to GiHub Notebook:
data-final-project/EDA with Visualization Lab.ipynb at main · SixtoM/data-final-project (github.com))

EXPLORATORY DATA ANALYSIS WITH VISUALIZATION TOOLS

3) CLASS VS ORBIT:



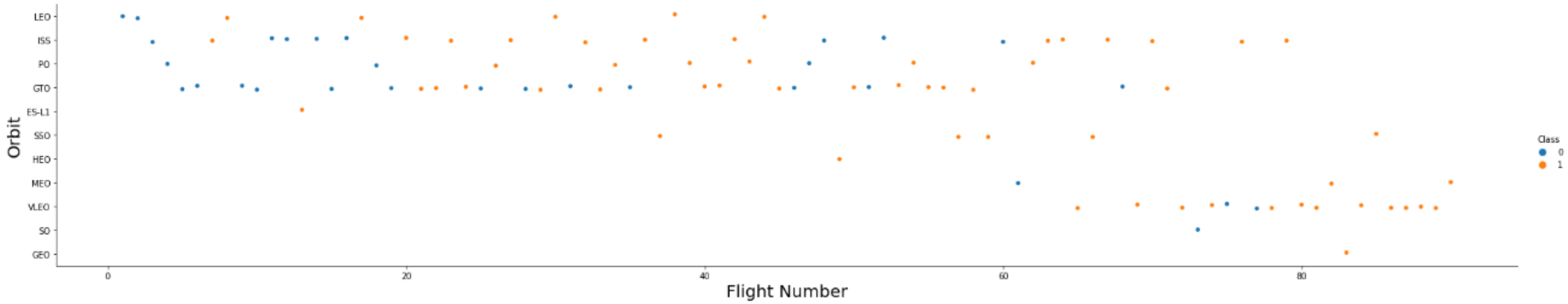
From this bar chart it can de deducted that the orbits that have the higher success rates are the following:

- **ES-L1**
- **GEP**
- **HEO**
- **SSO**

Link to GiHub Notebook:
data-final-project/EDA with Visualization Lab.ipynb at main · SixtoM/data-final-project (github.com))

EXPLORATORY DATA ANALYSIS WITH VISUALIZATION TOOLS

4) ORBIT VS FLIGHT NUMBERS:

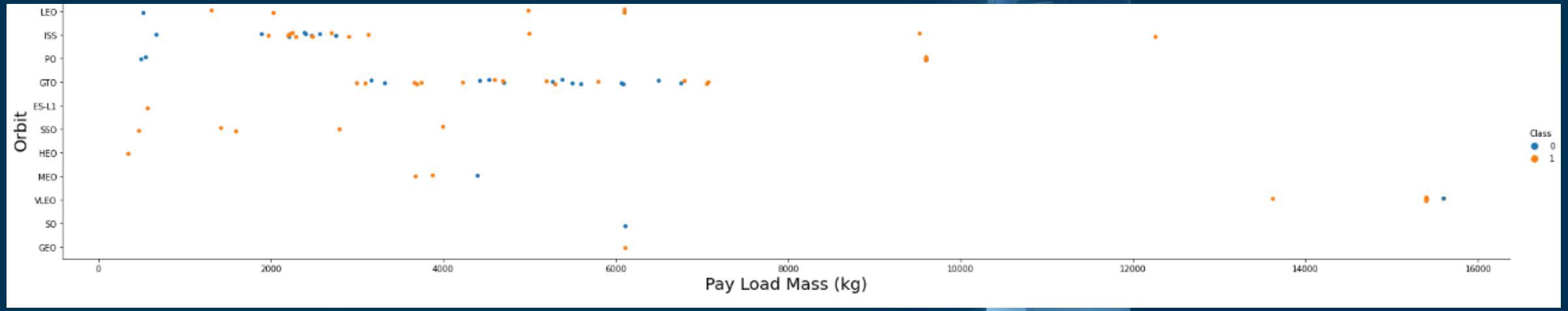


In this chart can be seen that the probability of a successful landing increases the numbers of flights in the LEO orbit as well as with ISS. However there seems to be a direct dependency between number of flights and successful type of rockets landing when other orbits are included, maybe with the exception of the ones previously mentioned and the flights in Orbit VLEO that only considers high quantity of flight numbers.

Link to GiHub Notebook:
data-final-project/EDA with Visualization Lab.ipynb at main · SixtoM/data-final-project (github.com))

EXPLORATORY DATA ANALYSIS WITH VISUALIZATION TOOLS

5) ORBIT VS PAY LOAD MASS:

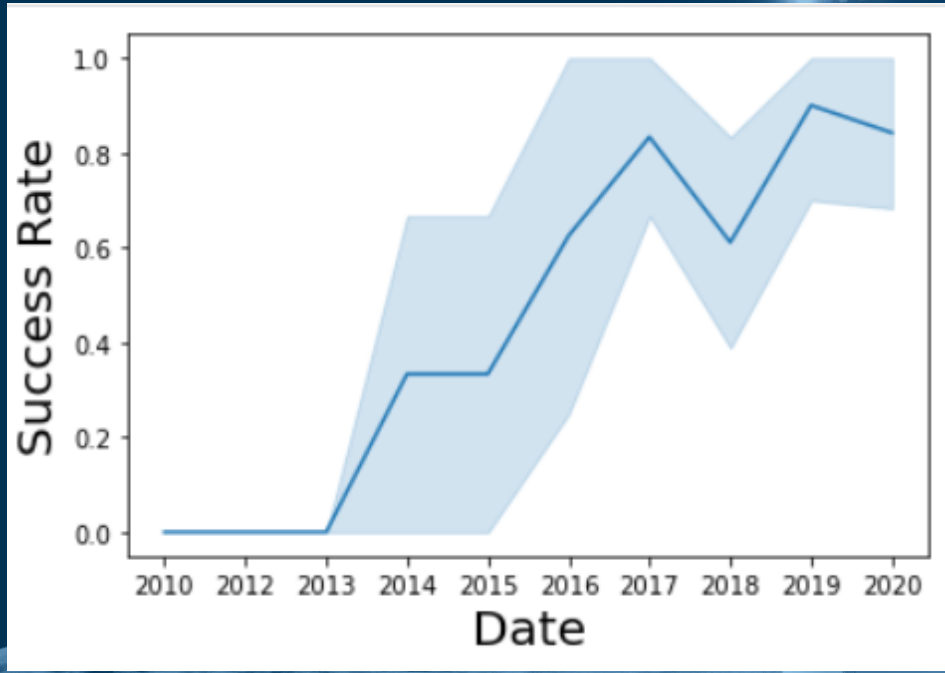


No significant relationship between the parameters of Pay Load Mass and Orbit were found with regards of the landing outcomes of the launches.
The only clear conclusion taken from this graph is that only successful landings were recorded from Orbits ISS, PO & LEO with the increase of the Pay Load Mass.

Link to GiHub Notebook:
data-final-project/EDA with Visualization Lab.ipynb at main · SixtoM/data-final-project (github.com))

EXPLORATORY DATA ANALYSIS WITH VISUALIZATION TOOLS

6) SUCCESS RATE VS YEAR:



From this graph is clear that there is an incremental relationship between the success rate of the rocket landings with the year they were launched; as the year increase so will the success rate.
The peak of highest successful rate was achieved in 2019 in approximately 90%.

Link to GiHub Notebook:
data-final-project/EDA with Visualization Lab.ipynb at main · SixtoM/data-final-project (github.com))

EXPLORATORY DATA ANALYSIS WITH SQL

To best understand, explore and analyze the data set, SQL queries were used for this purpose. The structure of the handle of the data with SQL was as follows:

1. CONNECT TO THE DATABASE

To load the SQL extension and established a connection with the database the following queries were used:

```
%load_ext sql
```

```
import csv, sqlite3

con = sqlite3.connect("my_data1.db")
cur = con.cursor()
```

```
!pip install -q pandas==1.1.5
```

```
%sql sqlite:///my_data1.db
```

```
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv")
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/pandas/core/generic.py:2605: UserWarning: The spaces in these column names will not be changed.
In pandas versions < 0.14, spaces were converted to underscores.
  sql.to_sql(
```

```
%sql PRAGMA table_info(SPACEXTBL)
```

Link to GiHub Notebook:
data-final-project/jupyter-labs-eda-sql-coursera_sqllite (1) (1).ipynb at main · SixtoM/data-final-project (github.com)

EXPLORATORY DATA ANALYSIS WITH SQL

2. EXPLORE THE DATABASE

- To display the name of the unique launch sites in the space mission, the following query was used

```
%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

- To display five (5) records where launch sites began with the string “CCA”, the following query was used:

```
%sql select * from SPACEXTBL where launch_site like 'CCA%' limit 20
```

```
 * 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

Link to GiHub Notebook:
data-final-project/jupyter-labs-eda-sql-coursera_sqllite (1) (1).ipynb at main · SixtoM/data-final-project (github.com)

EXPLORATORY DATA ANALYSIS WITH SQL

2. EXPLORE THE DATABASE

- To display the total pay load mass carried by the boosters, the following query was used:

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

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

sum(PAYLOAD_MASS__KG_)
45596

- To display the average pay load mass carried by booster version F9 v1.1, the following query was used:

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

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

avg(PAYLOAD_MASS__KG_)
2928.4

Link to GiHub Notebook:
data-final-project/jupyter-labs-eda-sql-coursera_sqllite (1) (1).ipynb at main · SixtoM/data-final-project (github.com)

EXPLORATORY DATA ANALYSIS WITH SQL

2. EXPLORE THE DATABASE

- To display the first successful landing date, the following query was used:

```
%sql select min(date) from SPACEXTBL where MISSION_OUTCOME ='Success'
```

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

min(date)
01-03-2013

- To display the list of boosters with successful landings and with 4,000 kg < Pay Load Mass < 6,000 kg, the following query was used:

```
%sql select (booster_version) from SPACEXTBL where MISSION_OUTCOME = 'Success' and PAYLOAD_MASS__KG_ between '4000' and '6000'
```

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

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Link to GiHub Notebook:
data-final-project/jupyter-labs-eda-sql-coursera_sqllite (1) (1).ipynb at main · SixtoM/data-final-project (github.com)

EXPLORATORY DATA ANALYSIS WITH SQL

2. EXPLORE THE DATABASE

- To display the list of total successful and failure landing missions, the following query was used:

```
%sql select MISSION_OUTCOME, count(*) from SPACEXTBL group by MISSION_OUTCOME
```

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

Mission_Outcome	count(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- To display the list of names of boosters that carried maximum payload mass, the following query was used:

```
%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

Link to GiHub Notebook:
data-final-project/jupyter-labs-eda-sql-coursera_sqllite (1) (1).ipynb at main · SixtoM/data-final-project (github.com)

EXPLORATORY DATA ANALYSIS WITH SQL

2. EXPLORE THE DATABASE

- To display the list of successful landing outcomes between dates 04/06/2010 and 20/03/2017 in descending order, the following query was used:

```
%sql select landing_outcome, Date from SPACEXTBL where (select date from SPACEXTBL order by substr(Date,7)||substr(Date, 4, 2)||substr(Date,7,4) DESC)
```

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

landing_outcome	Date
Success (ground pad)	22-12-2015
Success (ground pad)	18-07-2016
Success (ground pad)	19-02-2017
Success (ground pad)	01-05-2017
Success (ground pad)	03-06-2017
Success (ground pad)	14-08-2017
Success (ground pad)	07-09-2017
Success (ground pad)	15-12-2017
Success (ground pad)	08-01-2018

Link to GiHub Notebook:
data-final-project/jupyter-labs-eda-sql-coursera_sqllite (1) (1).ipynb at main · SixtoM/data-final-project (github.com)

VISUAL ANALYTICS USING FOLIUM MAPS

Understanding that the outcome of the booster's landing depends on the interaction of multiple factors, one good way of trying to determine the relationships between these parameters is by using visualization tools.
FOLIUM is a python map plotting library tool is used to visualize data in interactive maps. This tool was used to display the data collected for facilitate analysis.
The following elements were highlighted and marked in the map for data evaluation & analysis:

LAUNCH SITES

Marked all launch sites in map using Latitudes and Longitude coordinates by adding circles and name labels to each location.

LAUNCH OUTCOMES

Marked the type of landing outcomes from all launches covered in the data frame by using colored marker clusters.

DISTANCES

Signaled the distances between Launch sites and other locations like railways, highways and near cities.

Link to GiHub Notebook:
data-final-project/lab_jupyter_launch_site_location (1).py at main · SixtoM/data-final-project (github.com)

VISUAL ANALYTICS USING PLOTLY DASH

Using functions from Dash Plots are helpful for the generation of an interactive site where outputs are easily obtained from option of dropdown lists. In the Dash used for the analysis of the data collected, the following elements were developed:

LAUNCH SITES

A dropdown was enabled to select Launch site types

LAUNCH OUTCOMES

Used a pie chart to show the types of outcomes obtained when different or all launch sites have been selected

PAY LOADD MASS RANGE

A slider was added for the selection of Payload range as input

PAYLOAD MASS VS OUTCOME

Added a scatter chart to show the correlation between Payload and Launch success for different Booster Versions

PREDICTIVE ANALYSIS USING CLASSIFICATION MODELS

Link to GiHub Notebook:
data-final-project/Launch Prediction with Machine Learning.ipynb at main · SixtoM/data-final-project (github.com)

A NumPy was created from the column "**CLASS**" in the data frame

Data was standardized with ***StandardScaler***, fit and transformed

Data was split into training and testing sets with the ***train_test_split*** function

A *GridSearchCV* object with cv=10 was created to find the best parameters and applied these on *LogReg, SVM, Decision Tree and KNN models*

Accuracy was calculated using the test data with *method_score* for all prediction models

Develop and analyze the confusion matrix for all models

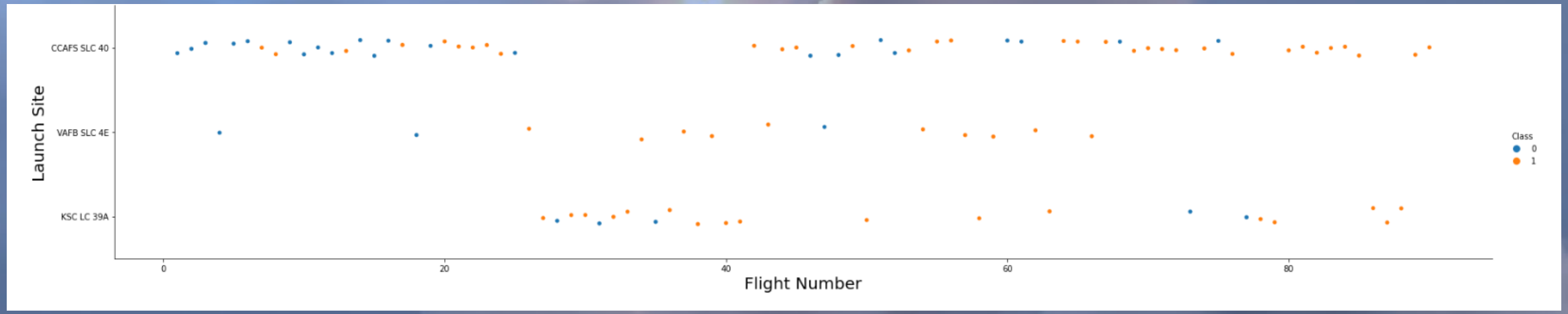
Evaluate the modes based on their accuracy scores and confusion matrix

Results

- Exploratory data analysis results
- Interactive Analytics
- Predictive Analysis results

EXPLORATORY DATA ANALYSIS RESULTS WITH VISUALIZATION

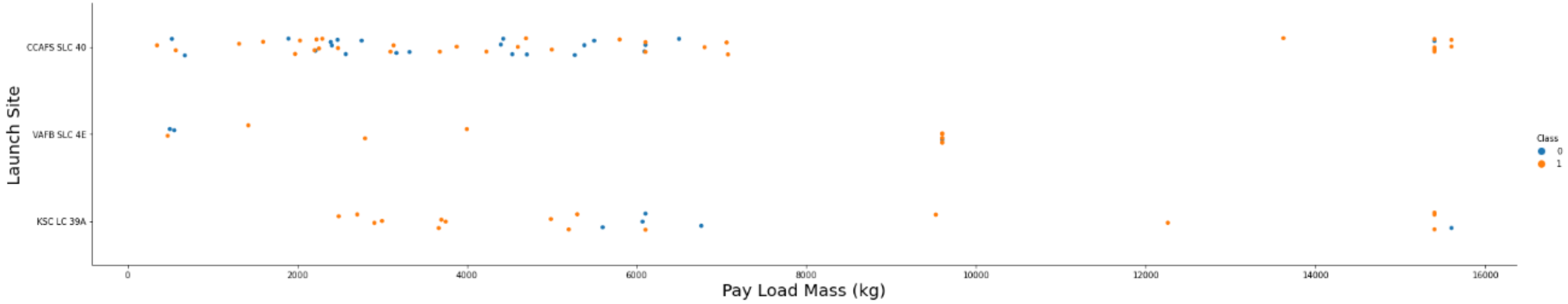
Flight Number vs Launch Site



- As a general result it is confirmed that as the number of flight increases so does the number of successful landings
- The probability of a successful landing per launch site seems to have the same tendency, however they do not share the same magnitude of number of landings
- When the number of flights are low, the probability of an unsuccessful landing are high

EXPLORATORY DATA ANALYSIS RESULTS WITH VISUALIZATION

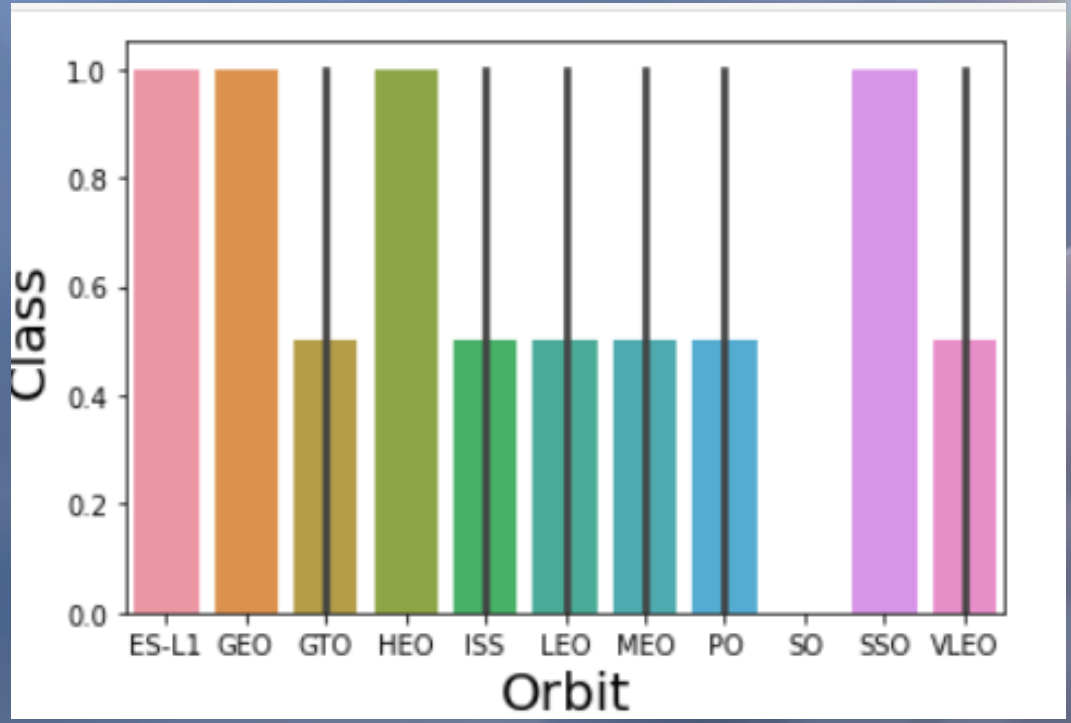
Payload vs Launch Site



- All launch sites showed higher probabilities of successful landings as payload mass increases
- CCAFS SLC 40 has a probability of over 80% when Payload mass is over 12,000 kg, but those odds drop down to 50% when the Payload mass is under 8,000 kg
- All launch sites have around 85% probability of a successful landing when pay load mass is over 8,000 kg
- No launches were made from VAFB SLC 4E with pay load masses over 10,000 kg

EXPLORATORY DATA ANALYSIS RESULTS WITH VISUALIZATION

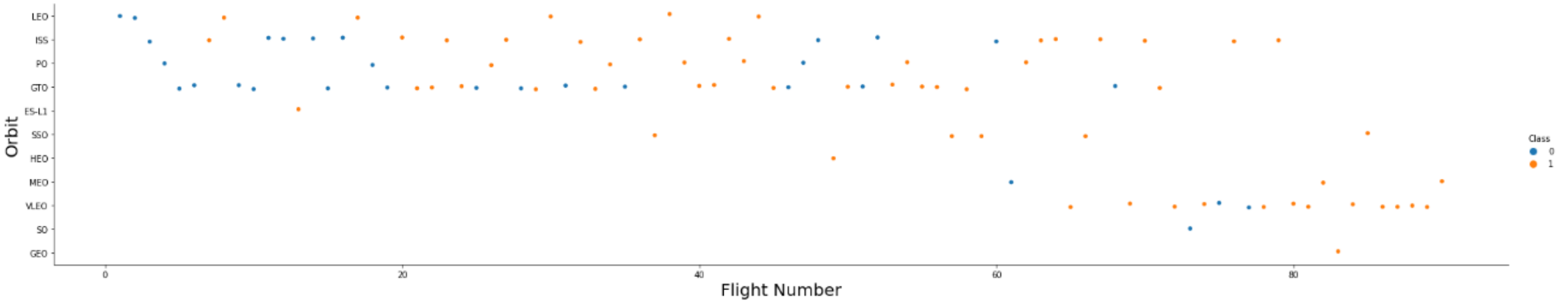
Success rate vs Orbit type



- Orbits with 100% success rate were ES-L1, GEO, HEO & SSO
- Orbits with 0% success rate was SO
- Orbits with 50% success rate were: GETO, ISS, LEO, MEO, PO & VLEO

EXPLORATORY DATA ANALYSIS RESULTS WITH VISUALIZATION

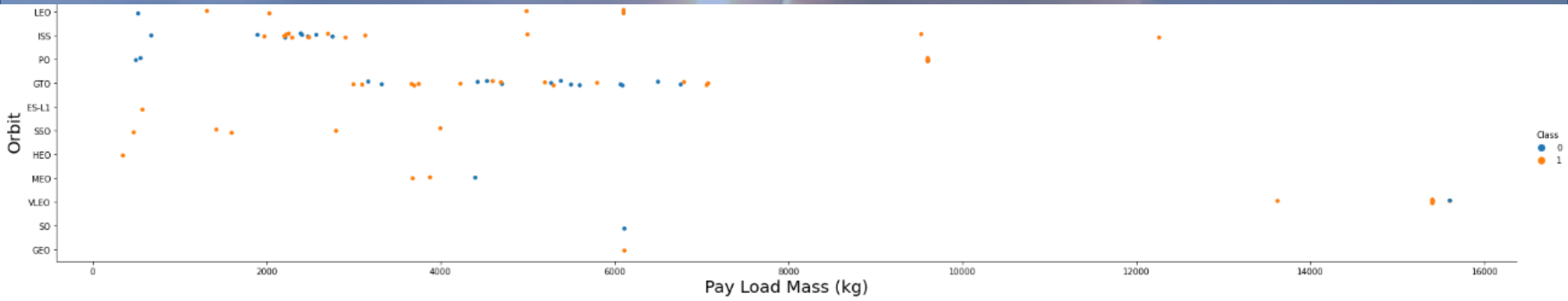
Flight number vs Orbit type



- Orbits LEO, ISS & VLEO show a clear tendency of having successful landing as the number of flights increases
- The rest of the orbits do not show a clear pattern or a linear relationship between these two parameters

EXPLORATORY DATA ANALYSIS RESULTS WITH VISUALIZATION

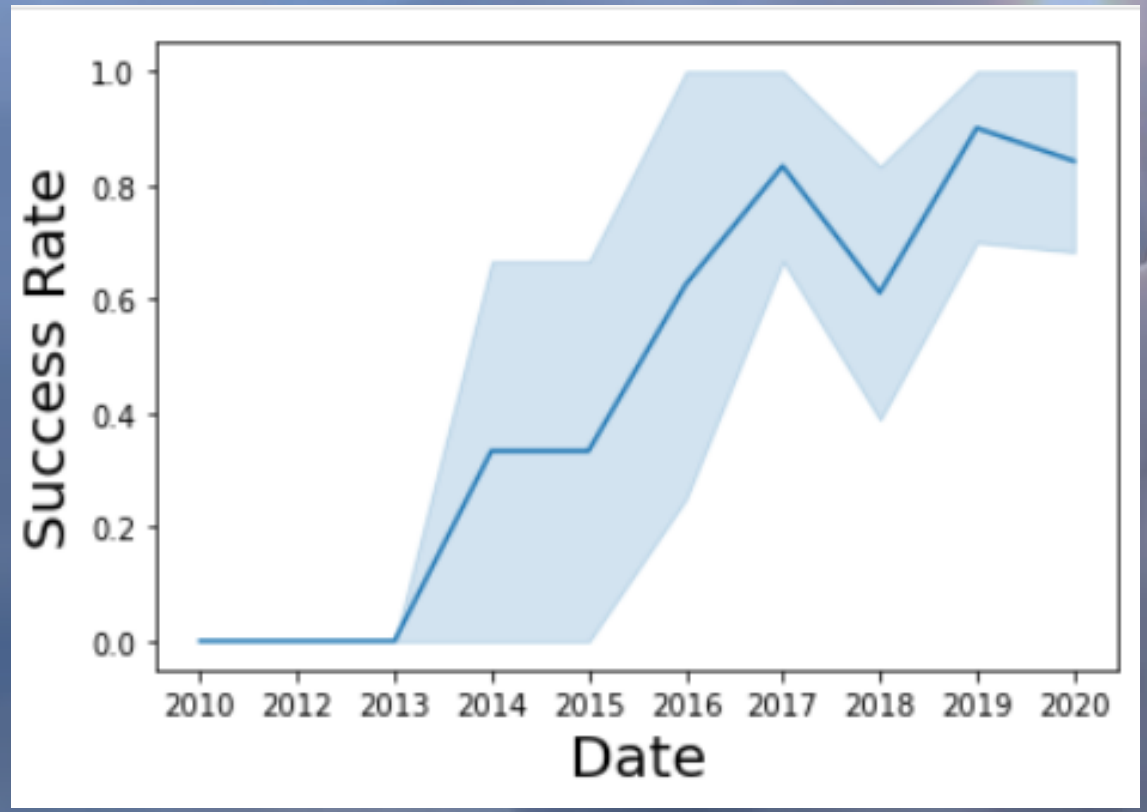
Payload Mass vs Orbit type



- Orbits LEO, ISS & PO show a tendency of obtaining successful landing as the Pay Load Mass increases
- Pay Load Mass does not seem to have a clear effect on the landing outcome when it comes to GTO orbit as the results are distributed
- With lower values of Pay Load Mass, Orbit SSO shows all successful landing, but no results are shown when higher values of Pay load mass are considered.

EXPLORATORY DATA ANALYSIS RESULTS WITH VISUALIZATION

Launch Success Yearly Trend



From this graph is clear that there is an incremental relationship between the success rate of the rocket landings with the year they were launched; as the year increase so will the success rate.

The peak of highest successful rate was achieved in 2019 in approximately 90%.

EXPLORATORY DATA ANALYSIS RESULTS WITH SQL

All Launch Site Names

```
In [13]: %sql select DISTINCT LAUNCH_SITE from SPACEXTBL

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

Out[13]:

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

Using SQL Magic we can query the database as shown. With this query we get the distinct entries from the column Launch_site in the table SPACE TBL which are all the launch sites pertinent to this study

Launch sites that begin with “CCA”

```
%sql select * from SPACEXTBL where launch_site like 'CCA%' limit 20
```

```
 * 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

By selecting “ * ” we request all columns of the data frame

The “where” clause is used to select the column to be evaluated with the “like” clause that specifies the condition of only having a “CCA” term

The limit clause restricts the number of the return

EXPLORATORY DATA ANALYSIS RESULTS WITH SQL

Total Payload Mass calculation

Display the total payload mass carried by boosters launched by NASA (CRS)

In [24]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where customer = 'NASA (CRS)'
```

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

Out[24]:

sum(PAYLOAD_MASS__KG_)
45596

The “Sum” clause calculates the total from the column specified inside the parenthesis

The “from” clause is to specify the table from where to retrieve the data and the “where” clause is used to identify the condition of the search

EXPLORATORY DATA ANALYSIS RESULTS WITH SQL

Average Payload Mass calculation

```
In [26]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where booster_version like 'F9 v1.1'

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

Out[26]:

avg(PAYLOAD_MASS__KG_)
2928.4

The “avg” clause calculates the average from the column specified inside the parenthesis

The “from” clause is to specify the table from where to retrieve the data, the “where” clause is used to identify the condition of the search and the “like” clause is used to determine the exact value of the condition

EXPLORATORY DATA ANALYSIS RESULTS WITH SQL

First Successful Landing Date

In [27]:
```
%sql select min(date) from SPACEXTBL where MISSION_OUTCOME ='Success'
```

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

Out[27]:

min(date)
01-03-2013

The “min” clause selects the minimum value from the column specified inside the parenthesis

The “from” clause is to specify the table from where to retrieve the data, the “where” clause is used to identify the condition of the search

EXPLORATORY DATA ANALYSIS RESULTS WITH SQL

List of Booster Names with successful landing with Payload mass between 4,000 and 6,000 kg

```
[29]: %sql select (booster_version) from SPACEXTBL where MISSION_OUTCOME = 'Success' and PAYLOAD_MASS__KG_ between '4000' and '6000'
```

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

The where clause restricts results to those with a payload mass between 4000 and 6000 kgs adding the “add” clause to add a second condition as “Mission_Outcome = Success”

EXPLORATORY DATA ANALYSIS RESULTS WITH SQL

Number of Outcome Types

In [30]:
```
%sql select MISSION_OUTCOME, count(*) from SPACEXTBL group by MISSION_OUTCOME
```

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

Out[30]:

Mission_Outcome	count(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Select Mission Outcomes and count of each type of Mission outcome as the Total Group by clause ensures we count the success and failures as separate groups

Boosters with maximum Payload Mass

In [32]:
```
%sql select (booster_version) from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

Out[32]:

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

- Select Unique Booster Versions to ensure we get distinct results
- Returned Payload Mass to see what the Max Payload is
- Where clause checks that the payload mass equals the max payload which is found via a sub query

EXPLORATORY DATA ANALYSIS RESULTS WITH SQL

Successful landings between dates

In [104…

```
%sql select landing_outcome, Date from SPACEXTBL where (select date from SPACEXTBL order by substr(Date,7)||substr(Date, 4, 2)||substr(Date,7,4) DESC)
```

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

Out[104…

landing_outcome	Date
Success (ground pad)	22-12-2015
Success (ground pad)	18-07-2016
Success (ground pad)	19-02-2017
Success (ground pad)	01-05-2017
Success (ground pad)	03-06-2017
Success (ground pad)	14-08-2017
Success (ground pad)	07-09-2017
Success (ground pad)	15-12-2017
Success (ground pad)	08-01-2018

The count of landing outcomes between dates June 4th, 2010, and March 20th, 2017 in descending order

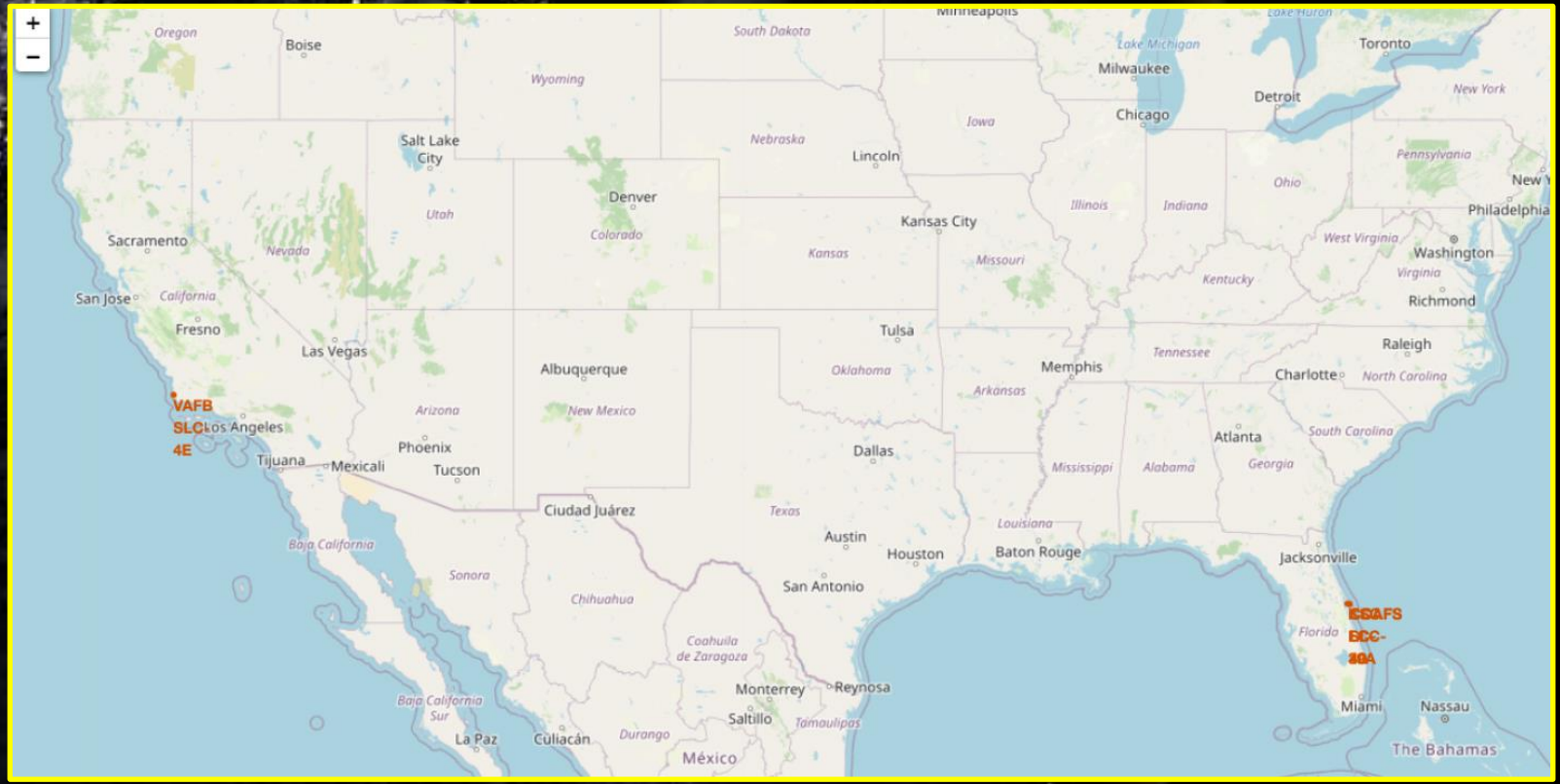
INTERACTIVE ANALYTICS

All launch sites from Space X are in the United States for a total of four with the following distribution:

- One (1) on the West Coast of California
- Three (3) on the East Coast of Florida

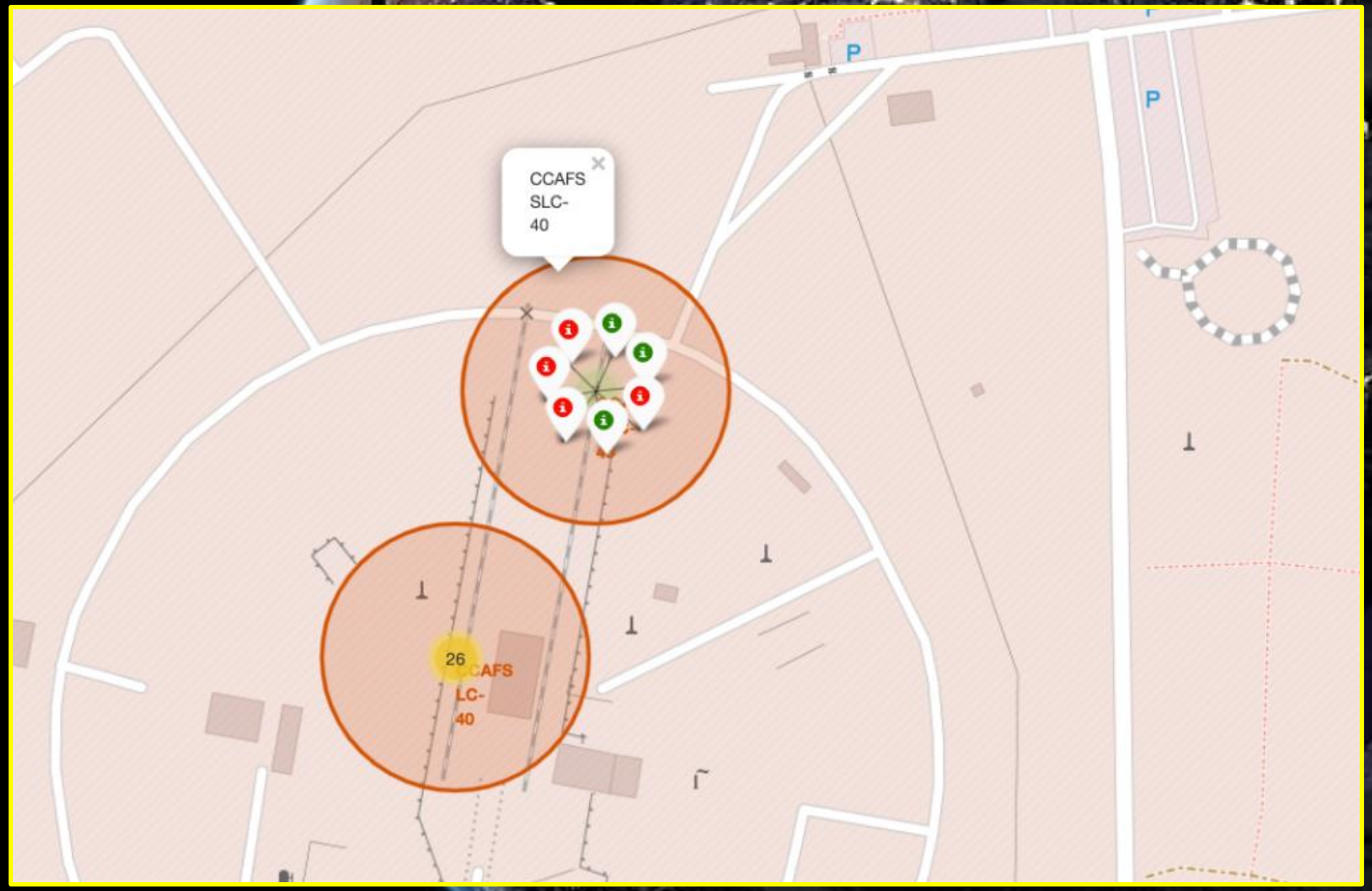
The reason of this distribution is that by being close to the ocean the probability of launch impacting civilians and buildings minimizes

SPACEX LAUNCH SITES GLOBAL MAP



INTERACTIVE ANALYTICS

CCAFS SLC-40 LAUNCH SITE

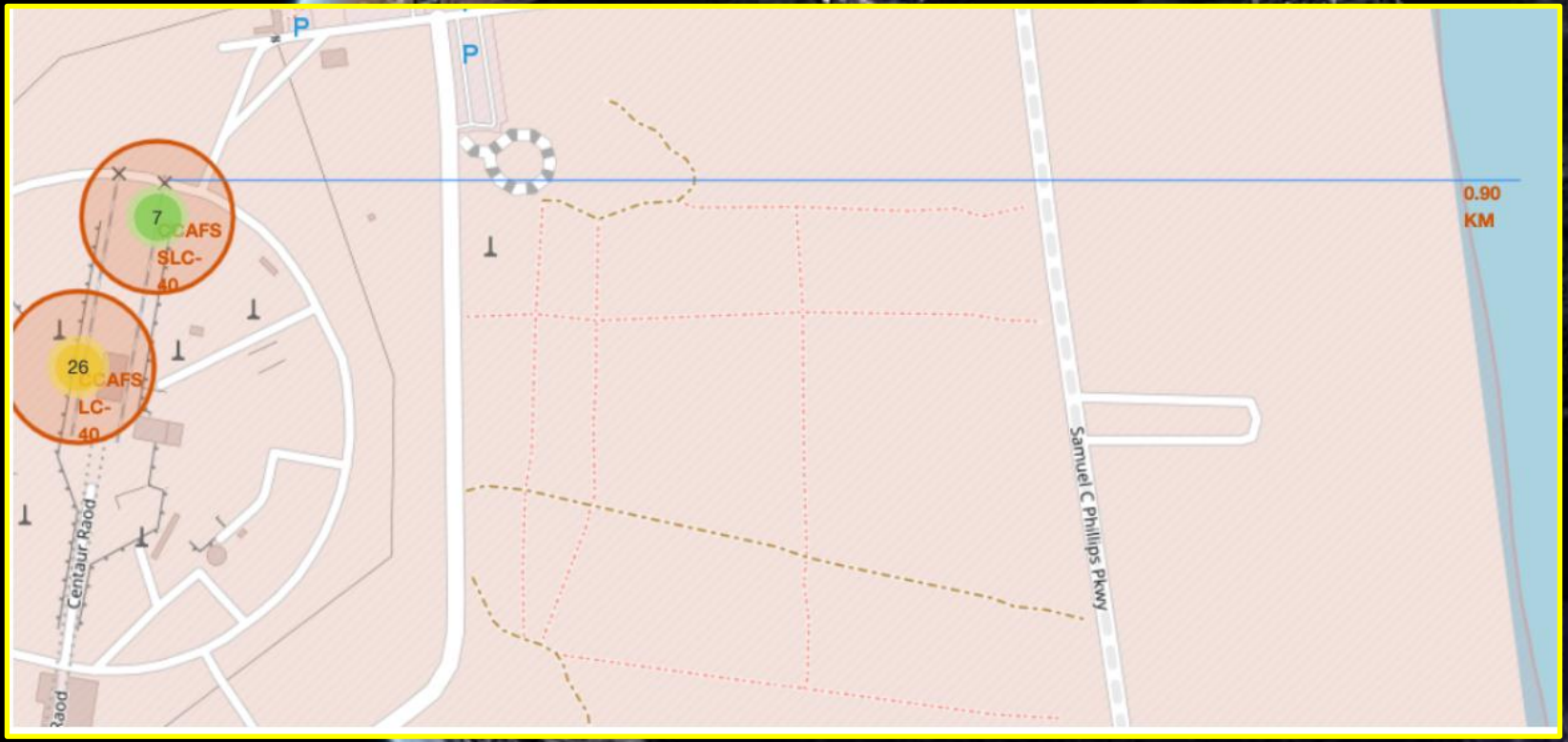


This map shows the successful (green tag) and failed (red tag) landings from CCAFS SLC-40 Launch Site.

In this launch site it was recorded four (4) failed landings vs only three (3) successful landings.

INTERACTIVE ANALYTICS

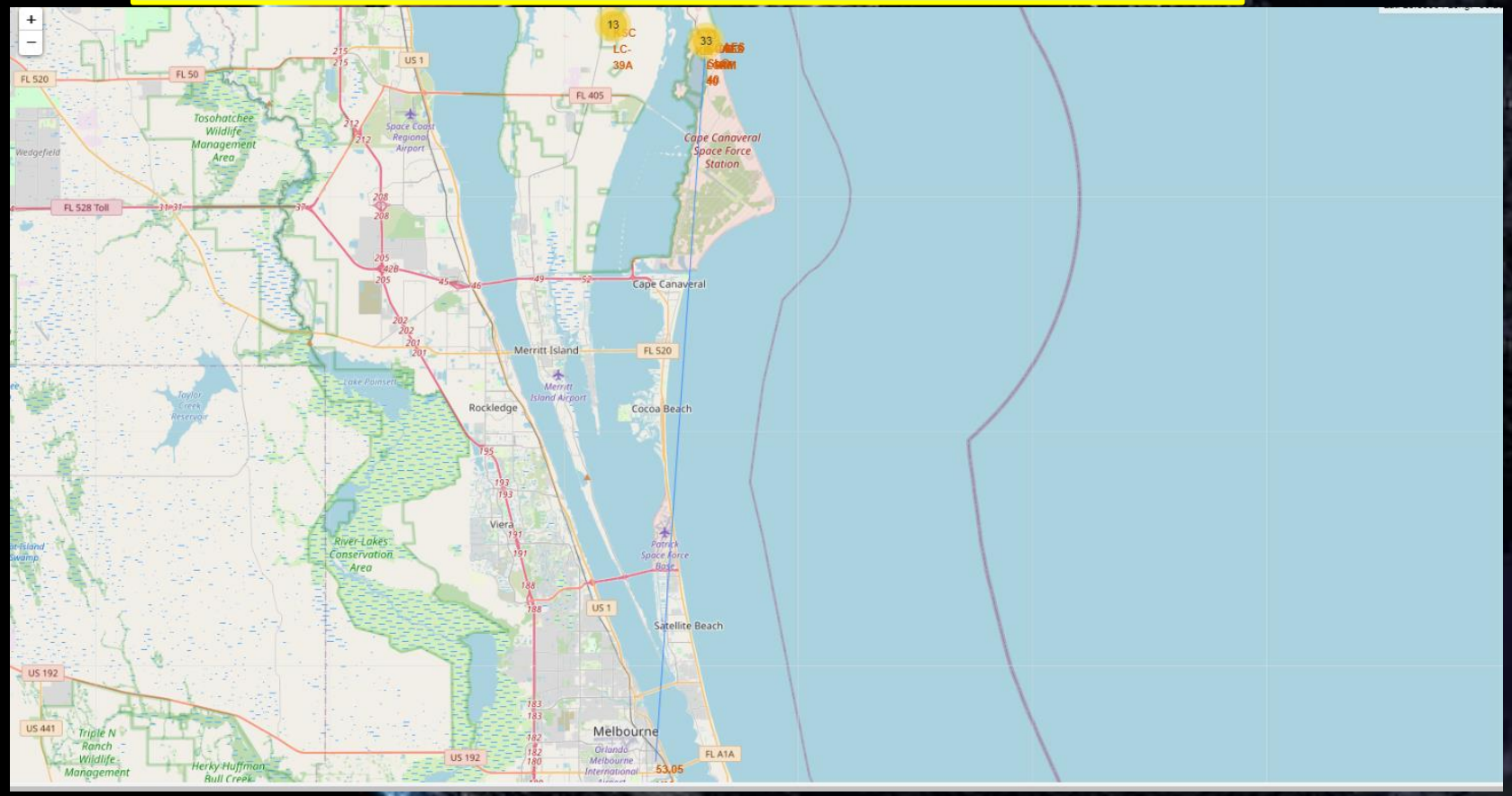
CCAFS SLC-40 LAUNCH SITE INFRAESTRUCTURE



This map shows that the nearest coastline to Cape Canaveral sites is within 1 km (~0.90 km) from CCAFS SLC-40

INTERACTIVE ANALYTICS

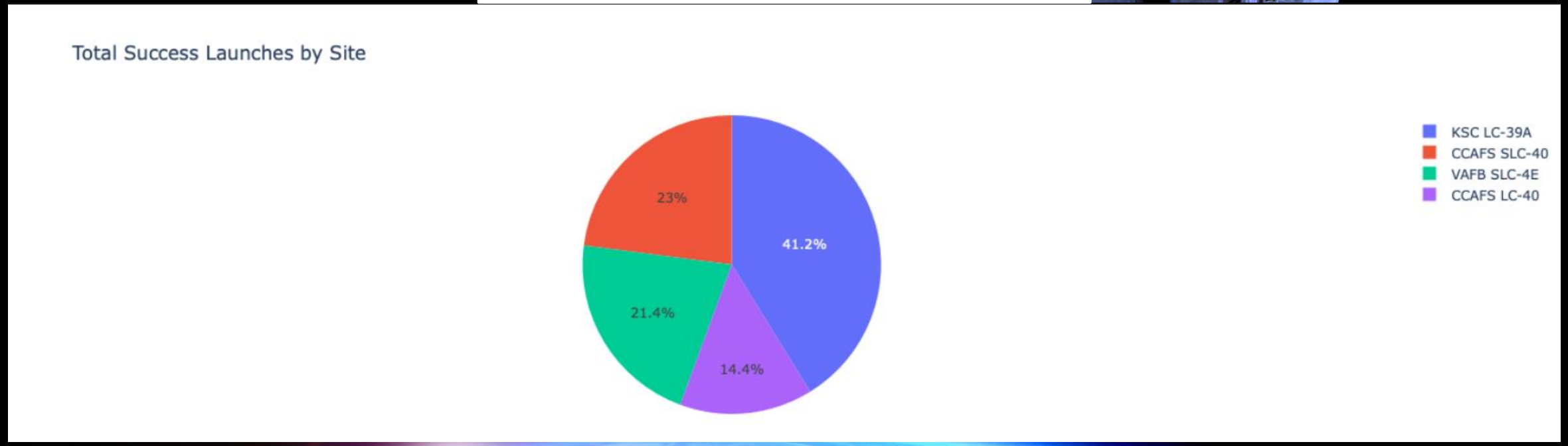
Distance from CAAFS SLC-40 to nearest city



The nearest populated city from CAAFS SLC-40 seems to be Melbourne which is located at 53 km.

INTERACTIVE ANALYTICS WITH DASHBOARDS

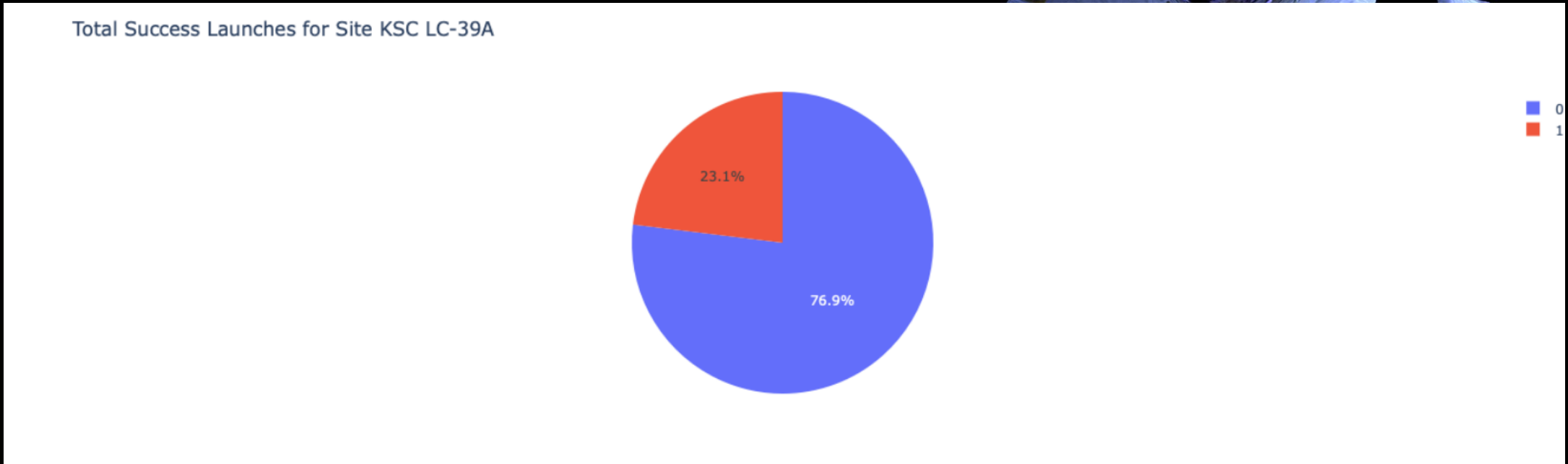
SITES LAUNCH SUCCESS



This chart shows the percentage distribution of rocket launches among all sites, being KSC LC-39A the most successful with 41% and CCAFS LC-40 the least successful with 14.4%.

INTERACTIVE ANALYTICS WITH DASHBOARDS

KSC LC-39A SITE LAUNCH SUCCESS



Knowing from previous chart that KSC LC-39A is the launch site with most successful landing, this chart confirms the reason behind as over 76% of the landings registered in this site were considered successful

INTERACTIVE ANALYTICS WITH DASHBOARDS

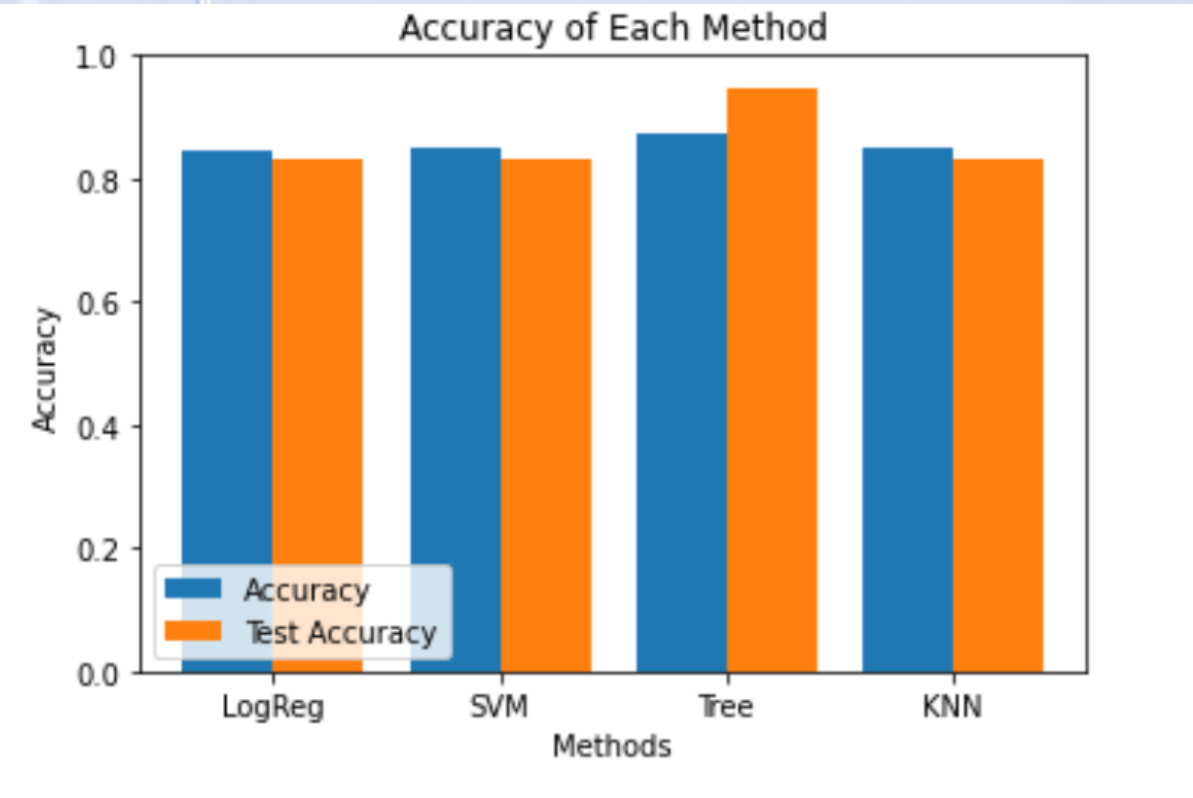
Payload Mass vs Launch Site

It can be deducted from these charts that the payload mass range with most probabilities of successful landings is between 2,000 kg and 5,550 kg

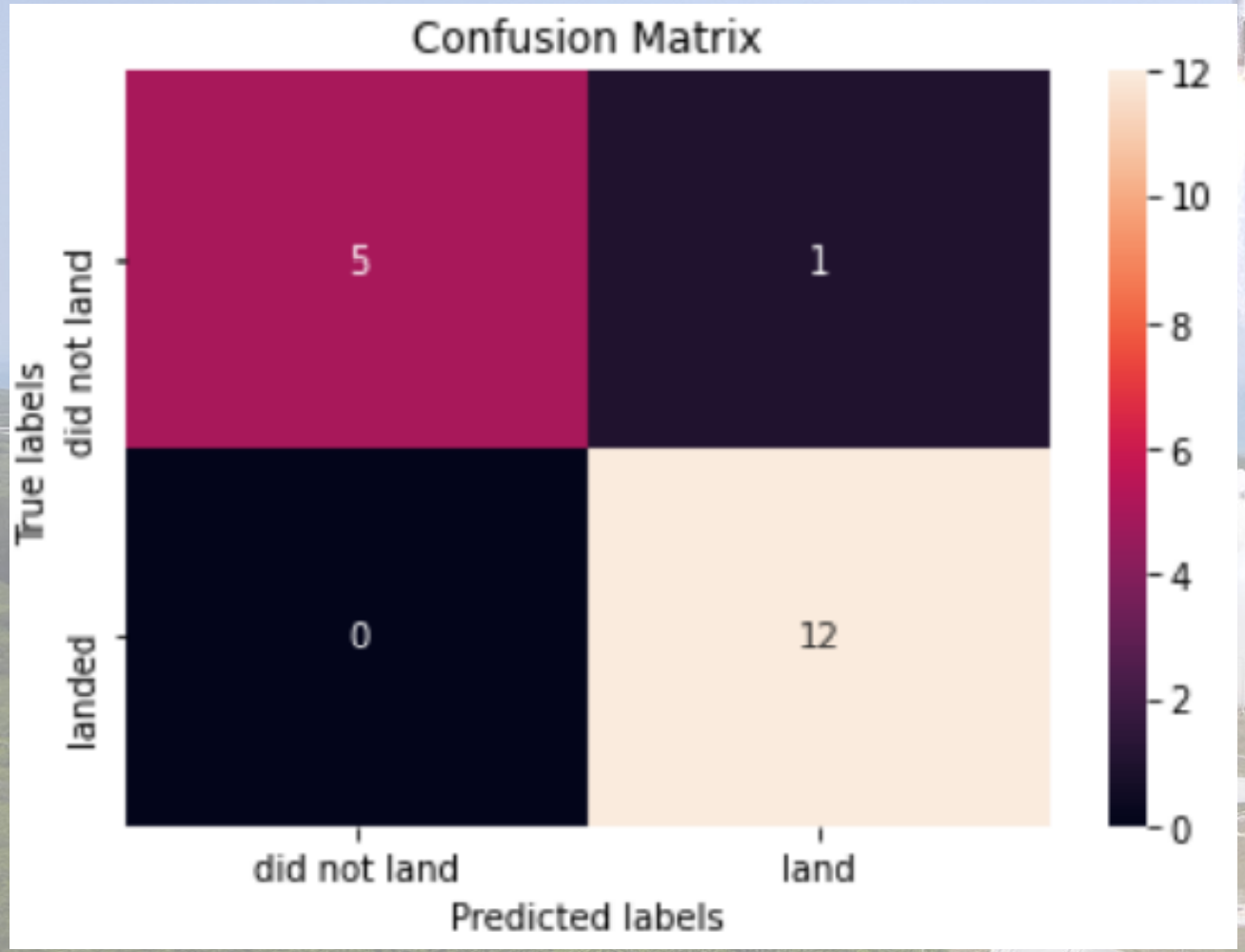
PREDICTIVE ANALYSIS RESULTS – CLASSIFICATION ACCURACY

As explained previously in other chapters of this presentation, four (4) models were used to predict launches. From all these models, The Decision Tree Classification model was proved to be the most accurate when using both with trained & test data in comparison with the other three models.
The Accuracy of the Decision Tree Model was proved to be 87.5% with the train data and 94.4% with the test data.

Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.875	0.94444
KNN	0.84821	0.83333



PREDICTIVE ANALYSIS RESULTS – TREE CONFUSION MATRIX



From this confusion matrix made for the Decision Tree Model and its predictions the following conclusions can be made:

- The model was only able to predict 5 unsuccessful landings out the 6. (sensitivity=83.3%)
- The model was able to predict that all twelve 12 rockets were going to land successfully. (Specificity=100%)
- The accuracy of the model is calculated like
  (5+12) / (5+1+0+12) = 94.44%

CONCLUSIONS

CONCLUSIONS

- There is a correlation between certain group of parameters like Payload Mass, Orbit, Launch site among others, that will be reflected in the landing outcome of the booster. Further studies are needed to understand the right specific weight of these parameters to efficiently predict successful landings and its associated cost.
- Orbits ES-L1, GEO, HEO and SSO have 100% success rate.
- Launch Sites are located close to coasts to prevent civilian fatalities and/or accidents.
- KSC LC-39A has the highest success rate of launches when compared to the rest of launch sites.
- Machine Learning algorithms can be used to predict based on collected data the outcome of a Falcon 9 launch by using trained & tested predictive models.
- Between SVM, KNN, Linear Regression and Decision Tree, the most accurate model based on the results of its predictions was the Decision Tree Classification Model.

APPENDIX

The GitHub repository containing the notebooks pertinent to the information displayed in this presentation can be found in the following URL:

SixtoM/data-final-project (github.com)

THANK YOU!