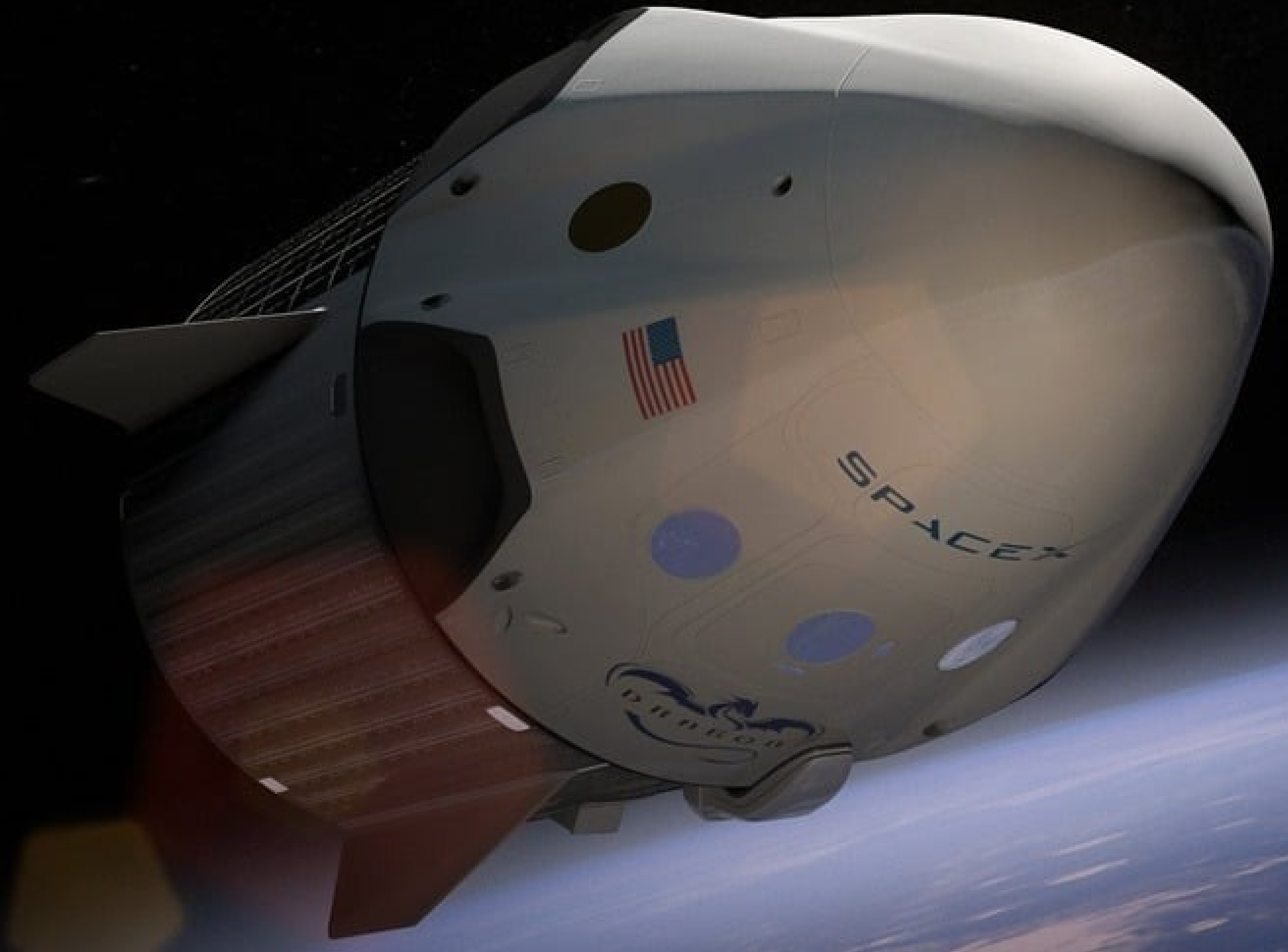
IBM Developer
SKILLS NETWORK

Data Science for the Space Race



Jorge Plazas
March 20 2023

Outline

- Executive Summary
- Introduction
- Part 1: Methodology and Data Wrangling
- Part 2: Exploratory Data Analysis
- Part 3: Launch Site Proximity Analysis
- Part 4: Machine Learning Models
- Conclusion
- Appendix

Executive Summary

In this project we apply a data science methodology framework in order to determine the probability of a successful landing of SpaceX's Falcon 9 first stage.

The results obtained can be used to asses the cost of a launch and provide other valuable information.

Introduction

Project background and context

SpaceX's Falcon 9 rocket launches are characterized by the re usability of their first stage. This characteristic was a large impact in reducing the cost of a rocket launch and as such gives SpaceX a large advantage over its competitors.

In this project we apply the data science methodology framework in order to assess the probability of a successful landing of SpaceX's Falcon 9 first stage.

The results obtained can be used to determine the cost of a launch, this information is specially valuable for a company that wants to bid against SpaceX for a rocket launch.

INTRODUCTION

The main questions our project answers are:

What is the probability of success of a SpaceX's Falcon 9 first stage landing?

What are the factors that determine whether a landing will be successful or not?

Part 1

Methodology

DATA COLLECTION
Sources and methodology

For the project we use historical data on SpaceX's Falcon 9 launches.
The data was collected from two primary sources:

- SpaceX data repository
The data was collected using a data collection API from SpaceXdata.

https://github.com/jorgeplazas/IBM-Applied-Data-Science-Capstone-Project/blob/0eba2a807281538b6f9f0f02d5a54e5ca7f17655/SpaceX_Data_Collection_Api.ipynb

- Wikipedia's article "List of Falcon 9 and Falcon Heavy launches"
The data was collected using webscraping with implemented with the BeautifulSoup

https://github.com/jorgeplazas/IBM-Applied-Data-Science-Capstone-Project/blob/8c312cd4cbd78c8442da4e54531c76b9d3e9dde5/SpaceX_Data_Collection_Webscraping.ipynb

DATA WRANGLING
Data cleaning and feature engineering

Once the data was collected we performed various data cleaning and data aggregation tasks. Null values were handled and categorical variables one-hot encoded. As a result we obtained a data frame ready for use. The code used for this part of the project is hosted at:

https://github.com/jorgeplazas/IBM-Applied-Data-Science-Capstone-Project/blob/15f05e023453378e2ec8f3eb210c985d81ce4350/SpaceX_Data_Wrangling.ipynb

ANALYSIS AND RESULTS

In the following slides we present the core of the analysis of this project in accordance to the remaining stages of the *Data Science Methodology* as it was used to address the above questions:

- An exploratory data analysis component.
- Analysis of geospacial data
- Machine learning algorithms

Part 2

Exploratory Data Analysis

First findings

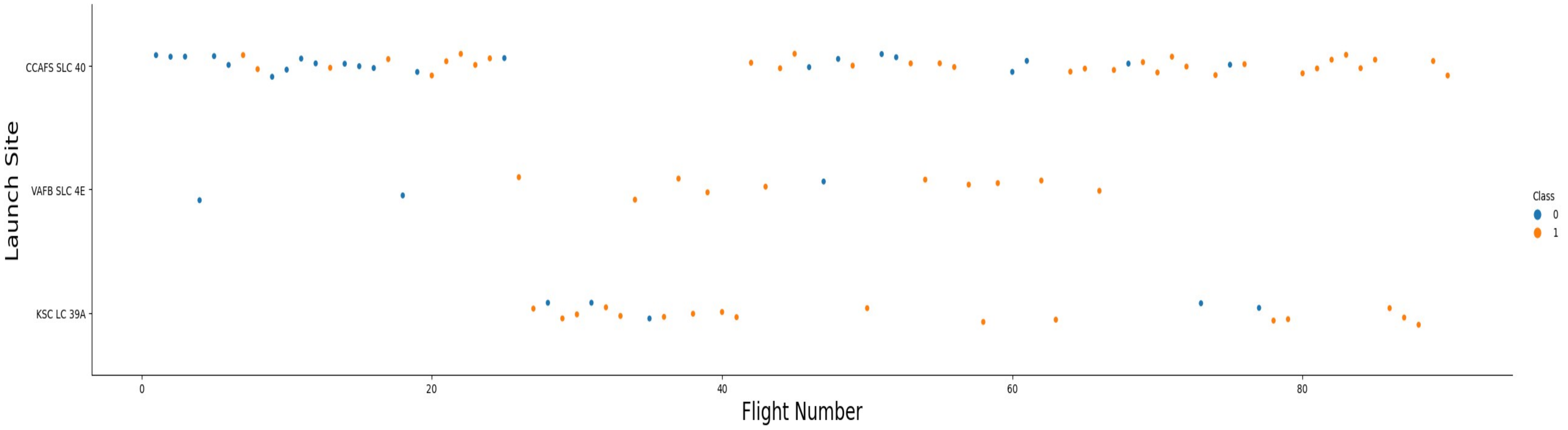
Exploratory Data Analysis I
Feature Correlation

In the following slides we present various plots which show the correlation between different features in our dataset.

In the following slides we present various plots which show the correlation between different features in our dataset.

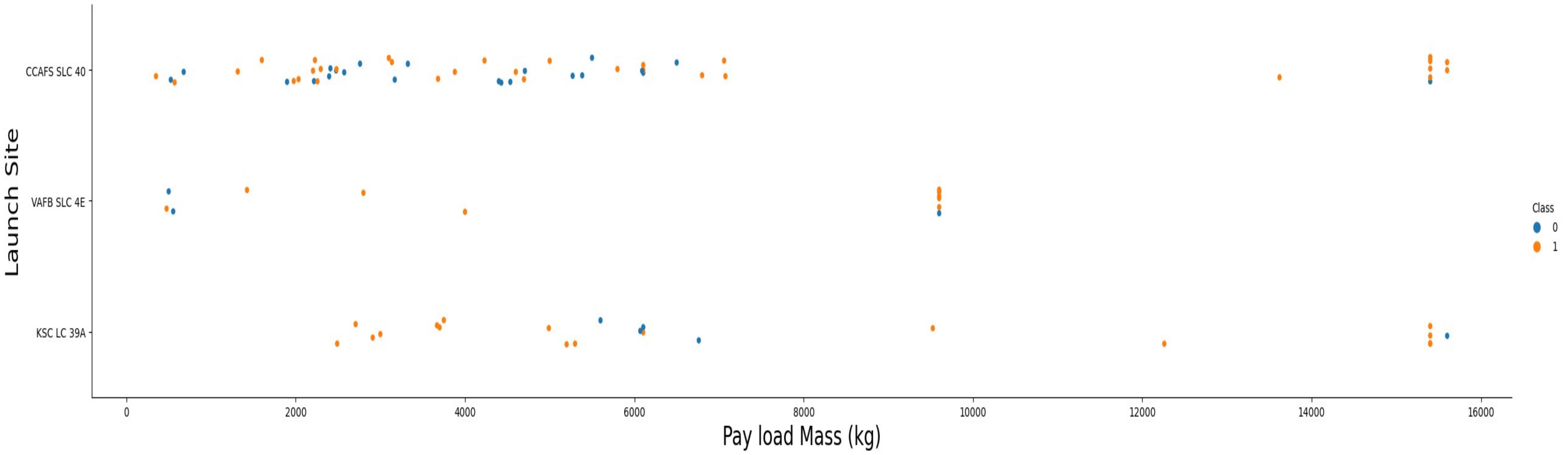
Exploratory Data Analysis I
Feature Correlation

Flight Number vs. Launch Site



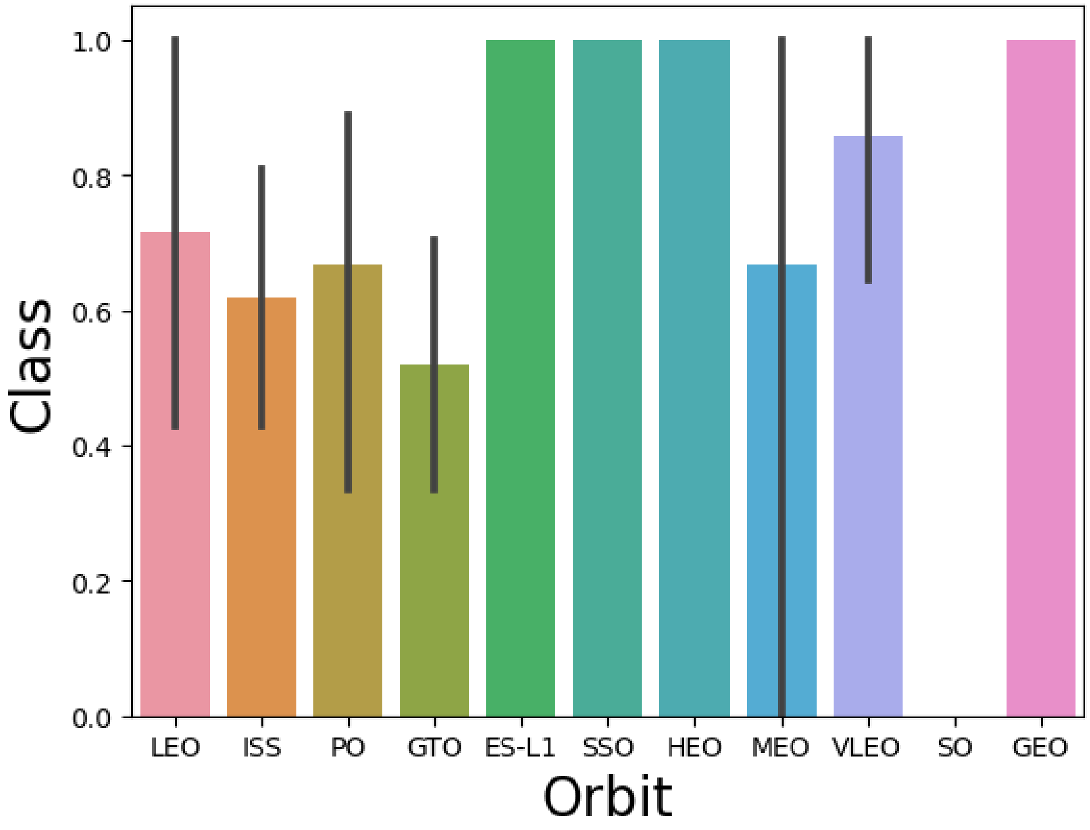
Exploratory Data Analysis I
Feature Correlation

Payload vs. Launch Site



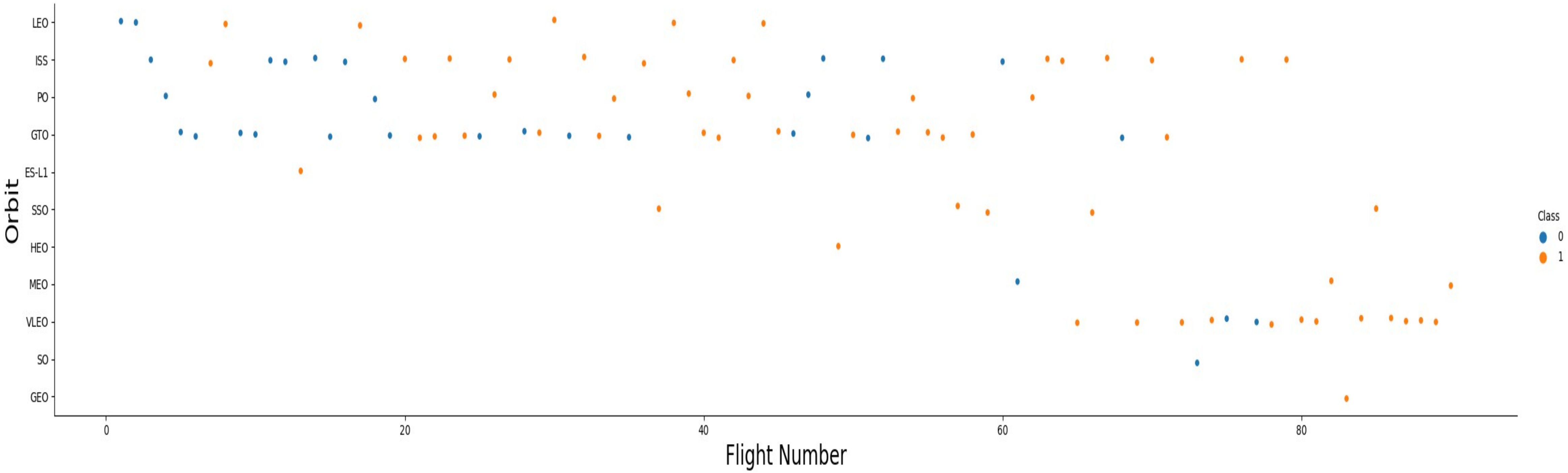
Exploratory Data Analysis I Feature Correlation

Success Rate vs. Orbit Type



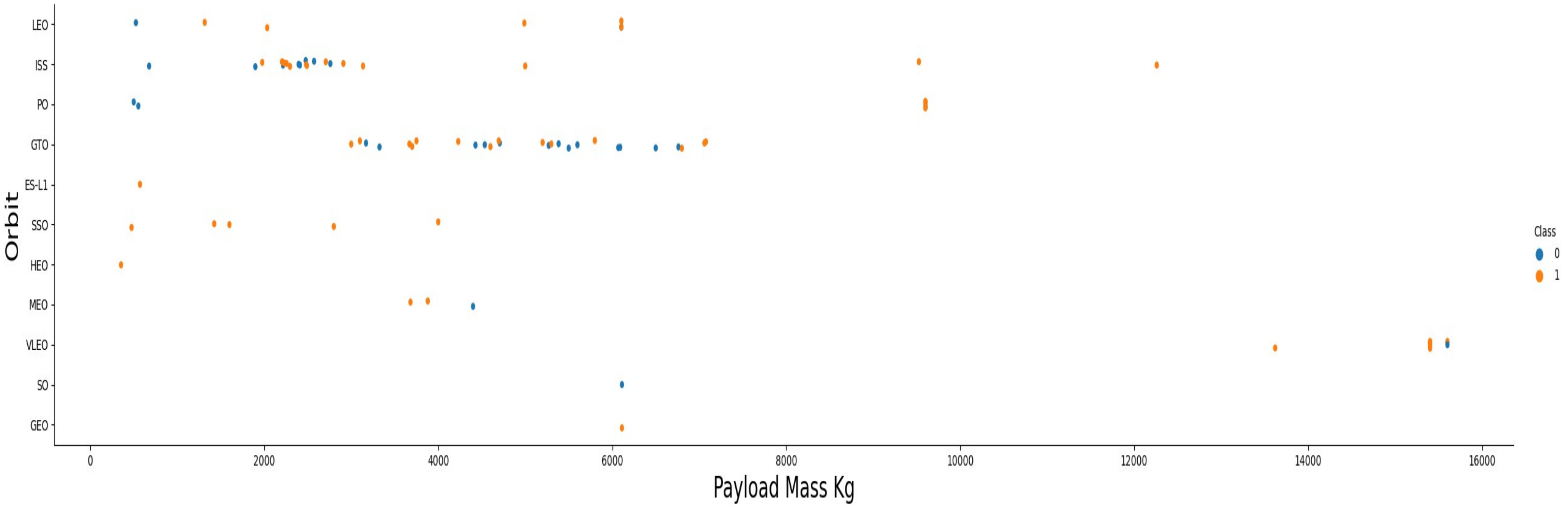
Exploratory Data Analysis I Feature Correlation

Flight Number vs. Orbit Type



Exploratory Data Analysis I
Feature Correlation

Payload vs. Orbit Type



Exploratory Data Analysis I
Feature Correlation

Code for this part of the project is hosted at:

https://github.com/jorgeplazas/
IBM-Applied-Data-Science-Capstone-Project/blob/
b7d05eb0472f7f532e52530a1bfbfdb9a60c4708/
SpaceX_EDA_Viz_and_FE.ipynb

Exploratory Data Analysis II
SQL query tasks

In this section we present the exploratory data analysis findings corresponding to tasks carried out using SQL.

The complete set of queries used has been included at the end of this presentation as part of the appendix.

Exploratory Data Analysis II
SQL query tasks

Unique launch sites in the space mission

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

Exploratory Data Analysis II
SQL query tasks

5 records where launch sites begin with the string 'CCA':

Row	Date	Time_UTC_	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS	Orbit	Customer
1	2013-12-03	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES
2	2014-01-06	22:06:00	F9 v1.1	CCAFS LC-40	Thaicom 6	3325	GTO	Thaicom
3	2014-08-05	08:00:00	F9 v1.1	CCAFS LC-40	AsiaSat 8	4535	GTO	AsiaSat
4	2014-09-07	05:00:00	F9 v1.1 B1011	CCAFS LC-40	AsiaSat 6	4428	GTO	AsiaSat
5	2015-03-02	03:50:00	F9 v1.1 B1014	CCAFS LC-40	ABS-3A Eutelsat 115 West B	4159	GTO	ABS Eutelsat

Exploratory Data Analysis II
SQL query tasks

Total payload mass carried by boosters launched by NASA (CRS):

111268

Exploratory Data Analysis II
SQL query tasks

Average payload mass carried by booster version F9 v1.1

2928.4

Exploratory Data Analysis II
SQL query tasks

Date when the first successful landing outcome in ground pad was achieved

2015-12-22

Exploratory Data Analysis II
SQL query tasks

Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000:

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

Exploratory Data Analysis II
SQL query tasks

Booster_versions which have carried the maximum payload mass (15600)

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

Exploratory Data Analysis II
SQL query tasks

Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

Row	month	Landing_Outcome	Booster_Version	Launch_Site
1	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Exploratory Data Analysis II
SQL query tasks

Code for this part of the project is hosted at:

https://github.com/jorgeplazas/IBM-Applied-Data-Science-Capstone-Project/blob/7f964862c5cb9fed67f188f4e3ed8a030db399e4/SpaceX_EDA_with_SQL.ipynb

Part 3

Launch Sites Proximity Analysis

Analysis of Geospacial Data

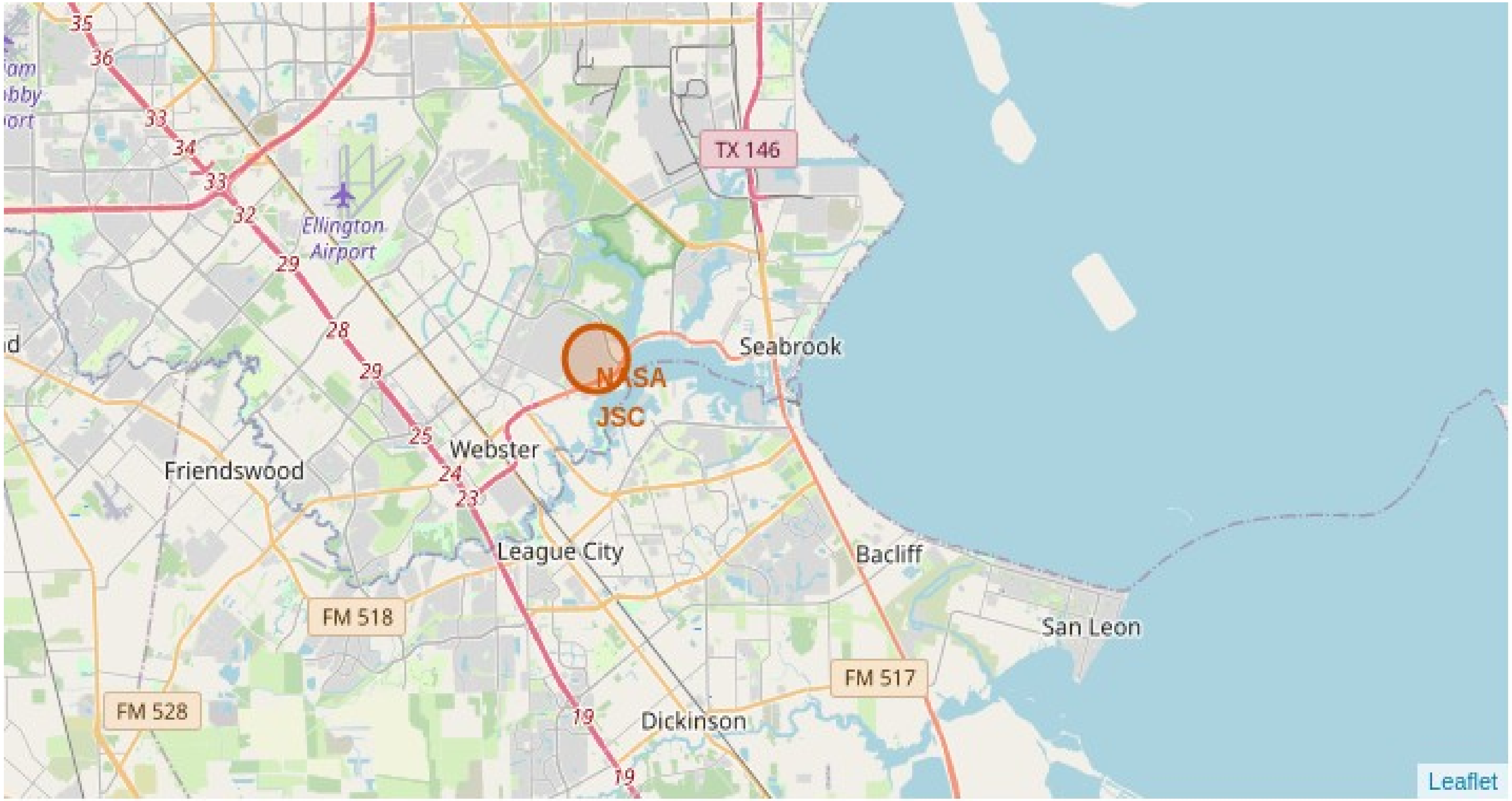
Analysis of Geospacial Data

Our previous analysis shows that success rate for landings differ according to the launch site.

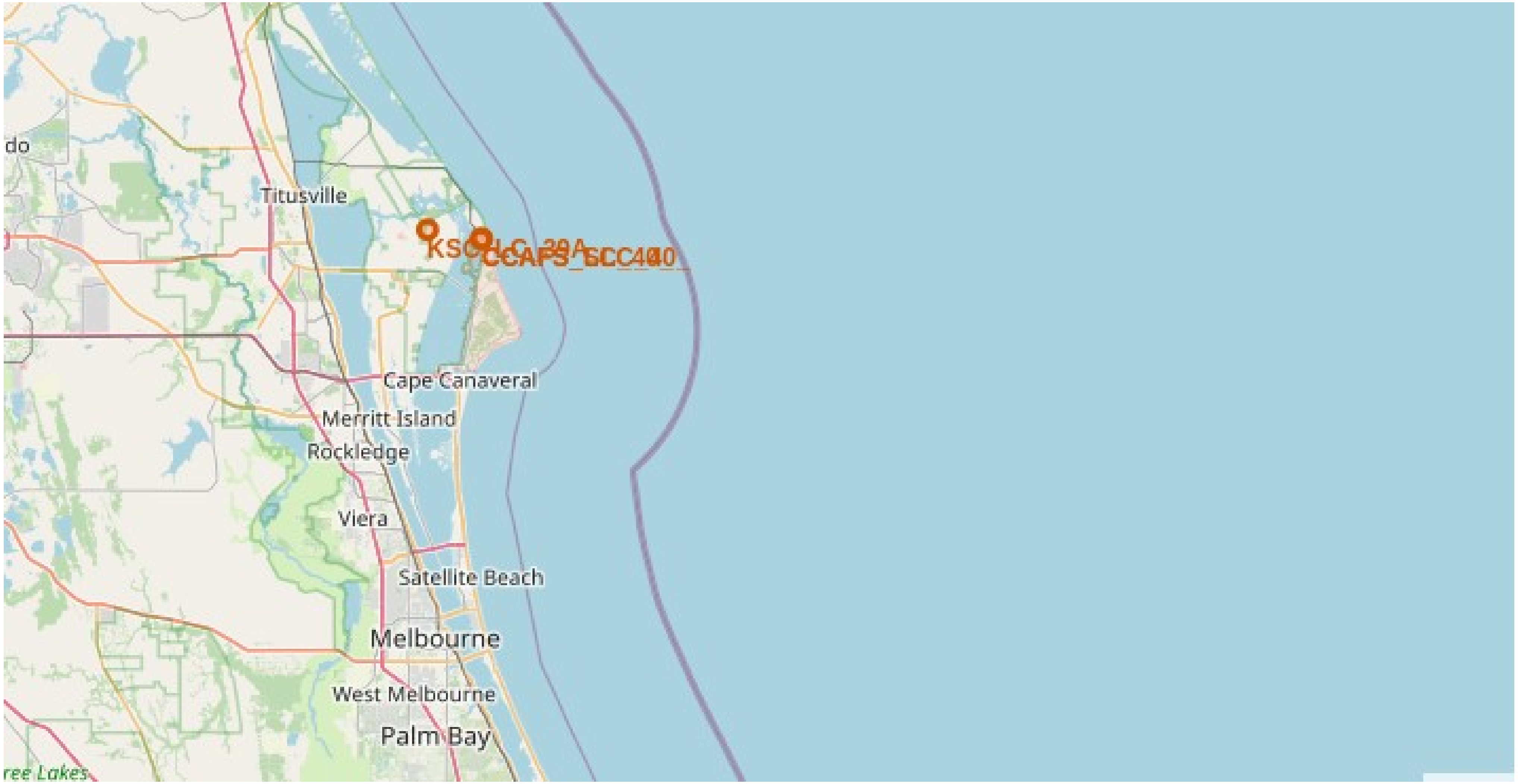
Due to the importance of geo-spacial data interactive maps were developed as part of the project using folium.

We use markers for each launch according to its location, distinguishing between successful and unsuccessful landings. The interactive maps can be used to determine proximity of these sites to roads, train tracks, coast lines and nearby towns.

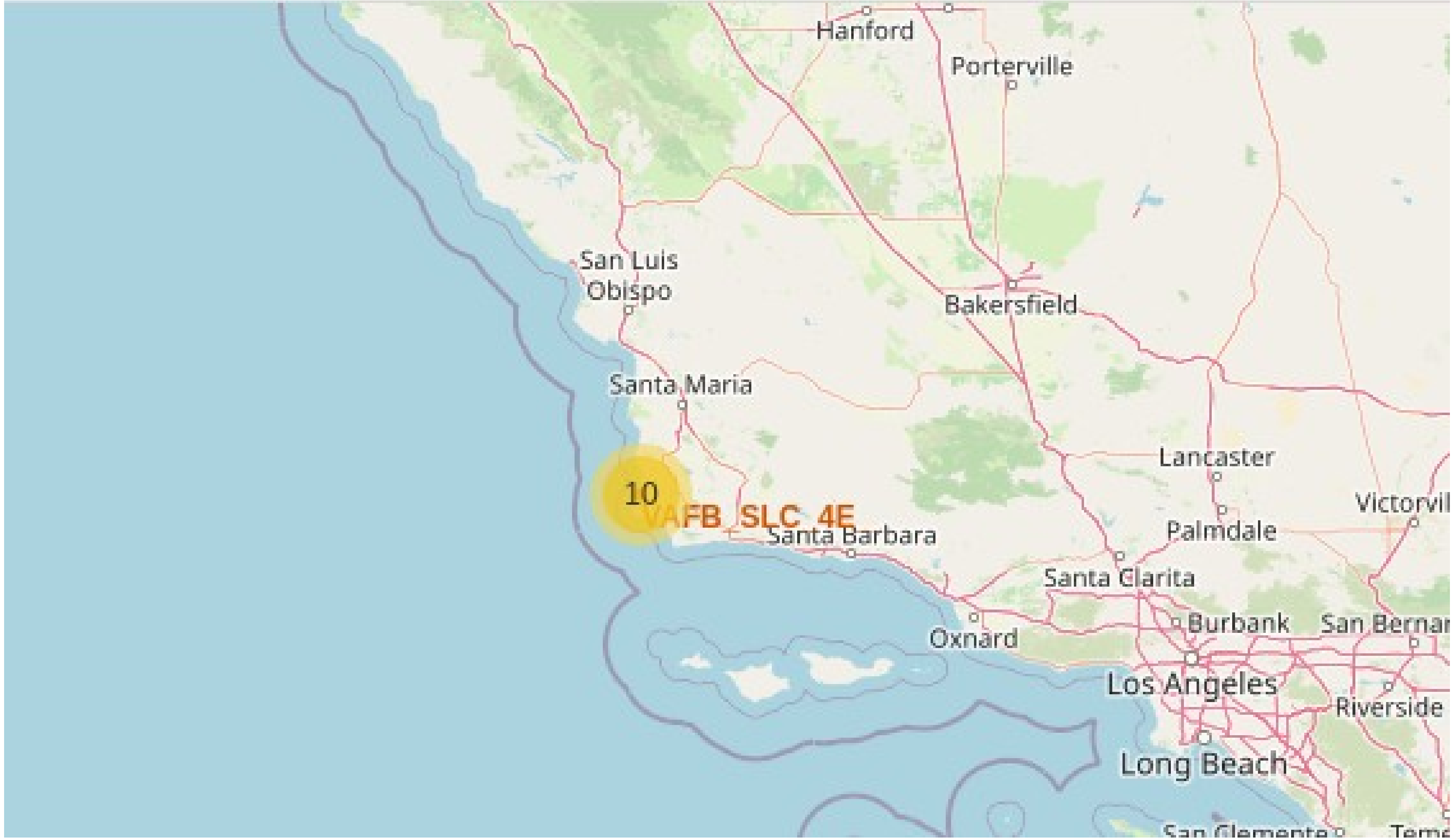
Analysis of Geospacial Data



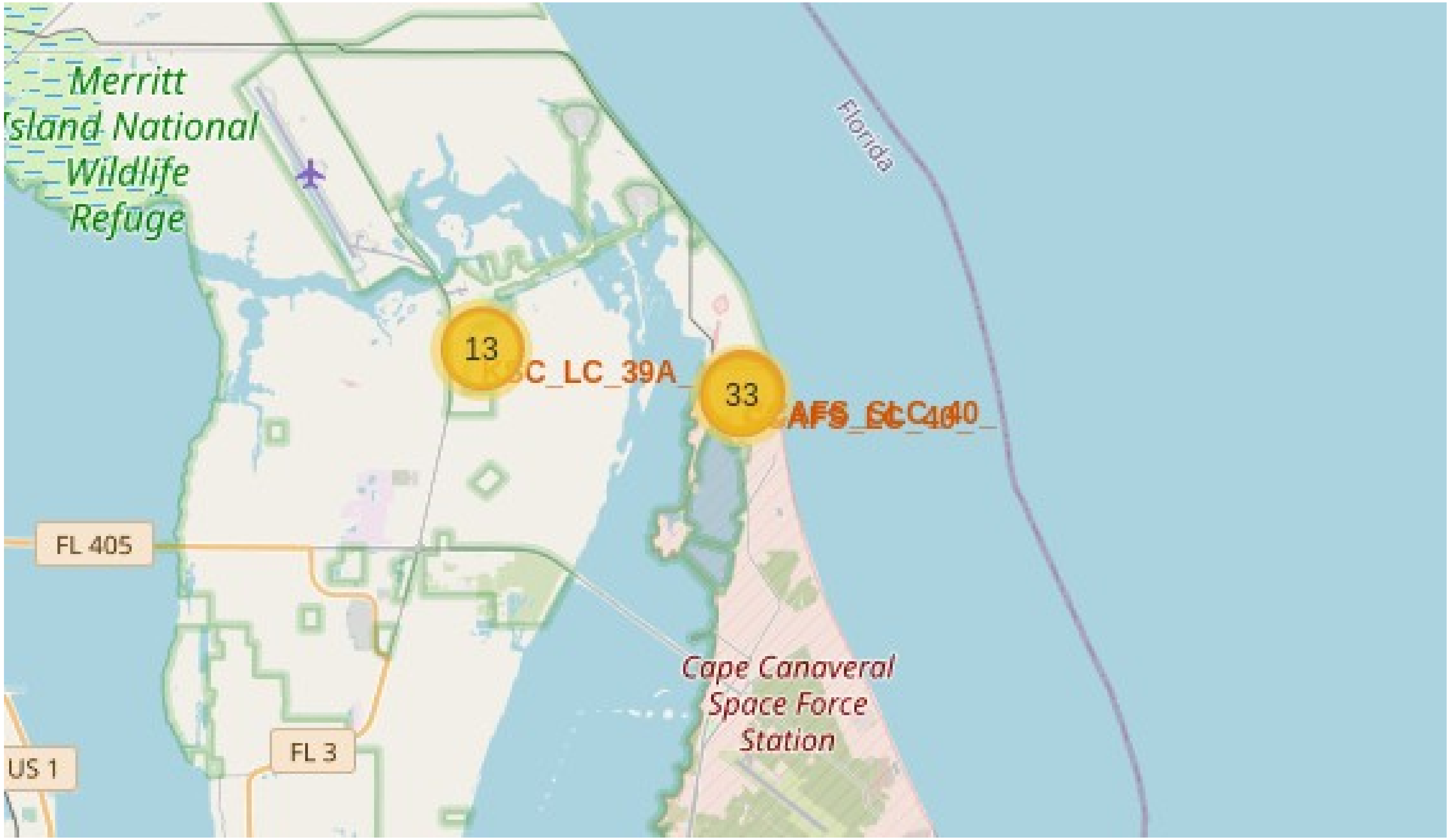
Analysis of Geospacial Data



Analysis of Geospacial Data



Analysis of Geospacial Data



Analysis of Geospacial Data

Analysis of Geospacial Data

Code for this part of the project is hosted at:

https://github.com/jorgeplazas/
IBM-Applied-Data-Science-Capstone-Project/blob/
b7d05eb0472f7f532e52530a1bfbfdb9a60c4708/
SpaceX_Launch_Site_Location.ipynb

Part 4
Predictive Analysis
Classification

SpaceX Falcon 9 first stage Landing Prediction
A Machine Learning approach

Understood as a machine learning task the prediction of whether a launch will have a successful or unsuccessful landing is an example of a

supervised binary classification task

As such it is natural to consider the performance of various classification models.

SpaceX Falcon 9 first stage Landing Prediction
A Machine Learning approach

In this project we compared the performance of the following models in the above task:

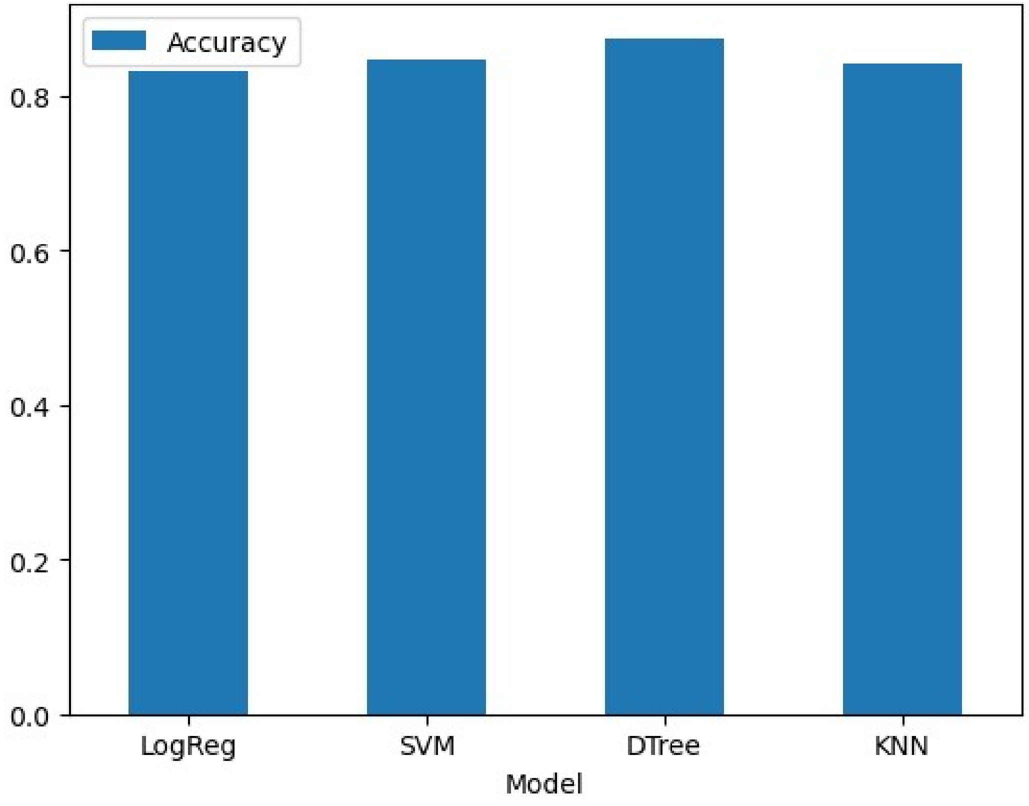
- Logistic regression (LogReg)
- Support vector machines (SVM)
- Decision tree classifier (DTree)
- K nearest neighbors (KNN)

For each of these models we split our data into two different sets for training and evaluation.

For each of the models a search was carried out in order to determine the optimal value of its hyperparameters.

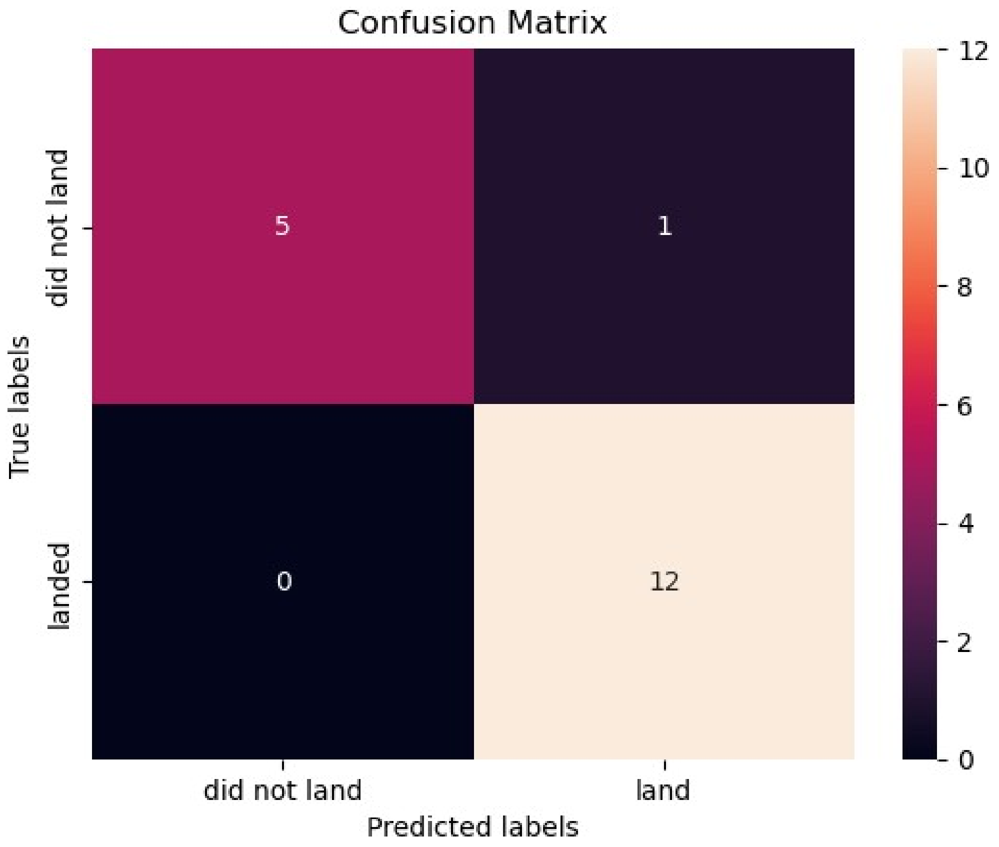
SpaceX Falcon 9 first stage Landing Prediction
A Machine Learning approach

The accuracy of each model was computed for the corresponding set of optimal parameters.



SpaceX Falcon 9 first stage Landing Prediction
A Machine Learning approach

The **Decision Tree Classifier** model has the highest accuracy. The corresponding confusion matrix is given by:



SpaceX Falcon 9 first stage Landing Prediction
A Machine Learning approach

Code for this part of the project is hosted at:

https://github.com/jorgeplazas/
IBM-Applied-Data-Science-Capstone-Project/blob/
b7d05eb0472f7f532e52530a1bfbfdb9a60c4708/
SpaceX_Machine_Learning_Prediction.ipynb

Part 5

Conclusions

Conclusions

1. Success rate for landings differ according to the launch site, the highest being those corresponding to KSC LC-39A and VAFB SLC 4E (77%). Due to the importance of geo-spacial data interactive maps were developed as part of the project.

2. For certain orbits (e.g. LEO) the success rate increases with the flight number. This indicates improvement over time for launches corresponding to these orbits.

3. For certain orbits (e.g. Polar) the success is dependent on payload mass. This indicates a dependence in the structural properties of the corresponding rockets.

4. A machine learning model that classifies successful/unsuccessful landings to high accuracy has been developed. The model is based on a decision tree classifier scheme and was was chosen above other classifier models by its performance.

Appendix

Appendix

Sample from the main Data Frame (final form):

	FlightNumber	PayloadMass	Flights	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	Orbit_GTO	Orbit_HEO	Orbit_ISS	...	Serial_B1058	Serial_B1059	Serial_B1060	S
0	1.0	6104.959412	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
1	2.0	525.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
2	3.0	677.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	
3	4.0	500.000000	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
4	5.0	3170.000000	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
85	86.0	15400.000000	2.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	
86	87.0	15400.000000	3.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	1.0	0.0	0.0	
87	88.0	15400.000000	6.0	5.0	5.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	
88	89.0	15400.000000	3.0	5.0	2.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	1.0	
89	90.0	3681.000000	1.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	

Dataframe Metadata

<class 'pandas.core.frame.DataFrame'>

- RangeIndex: 90 entries, 0 to 89
- Data columns (total 83 columns):
- #   Column   Non-Null Count Dtype
- --- ------   -------------- -----
- 0 FlightNumber   90 non-null   float64
- 1 PayloadMass   90 non-null   float64
- 2 Flights   90 non-null   float64
- 3 Block   90 non-null   float64
- 4 ReusedCount   90 non-null   float64
- 5 Orbit_ES-L1   90 non-null   float64
- 6 Orbit_GEO   90 non-null   float64
- 7 Orbit_GTO   90 non-null   float64
- 8 Orbit_HEO   90 non-null   float64
- 9 Orbit_ISS   90 non-null   float64
- 10 Orbit_LEO   90 non-null   float64
- 11 Orbit_MEO   90 non-null   float64
- 12 Orbit_PO   90 non-null   float64
- 13 Orbit_SO   90 non-null   float64
- 14 Orbit_SSO   90 non-null   float64
- 15 Orbit_VLEO   90 non-null   float64
- 16 LaunchSite_CCAFS SLC 40   90 non-null   float64
- 17 LaunchSite_KSC LC 39A   90 non-null   float64
- 18 LaunchSite_VAFB SLC 4E   90 non-null   float64
- 19 LandingPad_5e9e3032383ecb267a34e7c7 90 non-null   float64
- 20 LandingPad_5e9e3032383ecb554034e7c9 90 non-null   float64
- 21 LandingPad_5e9e3032383ecb6bb234e7ca 90 non-null   float64
- 22 LandingPad_5e9e3032383ecb761634e7cb 90 non-null   float64
- 23 LandingPad_5e9e3033383ecbb9e534e7cc 90 non-null   float64
- 24 Serial_B0003   90 non-null   float64
- 25 Serial_B0005   90 non-null   float64
- 26 Serial_B0007   90 non-null   float64
- 27 Serial_B1003   90 non-null   float64
- 28 Serial_B1004   90 non-null   float64
- 29 Serial_B1005   90 non-null   float64
- 30 Serial_B1006   90 non-null   float64
- 31 Serial_B1007   90 non-null   float64
- 32 Serial_B1008   90 non-null   float64
- 33 Serial_B1010   90 non-null   float64
- 34 Serial_B1011   90 non-null   float64
- 35 Serial_B1012   90 non-null   float64
- 36 Serial_B1013   90 non-null   float64
- 37 Serial_B1015   90 non-null   float64
- 38 Serial_B1016   90 non-null   float64
- 39 Serial_B1017   90 non-null   float64
- 40 Serial_B1018   90 non-null   float64
- 41 Serial_B1019   90 non-null   float64
- 42 Serial_B1020   90 non-null   float64
- 43 Serial_B1021   90 non-null   float64
- 44 Serial_B1022   90 non-null   float64
- 45 Serial_B1023   90 non-null   float64
- 46 Serial_B1025   90 non-null   float64
- 47 Serial_B1026   90 non-null   float64
- 48 Serial_B1028   90 non-null   float64
- 49 Serial_B1029   90 non-null   float64
- 50 Serial_B1030   90 non-null   float64
- 51 Serial_B1031   90 non-null   float64
- 52 Serial_B1032   90 non-null   float64
- 53 Serial_B1034   90 non-null   float64
- 54 Serial_B1035   90 non-null   float64
- 55 Serial_B1036   90 non-null   float64
- 56 Serial_B1037   90 non-null   float64
- 57 Serial_B1038   90 non-null   float64
- 58 Serial_B1039   90 non-null   float64
- 59 Serial_B1040   90 non-null   float64
- 60 Serial_B1041   90 non-null   float64
- 61 Serial_B1042   90 non-null   float64
- 62 Serial_B1043   90 non-null   float64
- 63 Serial_B1044   90 non-null   float64
- 64 Serial_B1045   90 non-null   float64
- 65 Serial_B1046   90 non-null   float64
- 66 Serial_B1047   90 non-null   float64
- 67 Serial_B1048   90 non-null   float64
- 68 Serial_B1049   90 non-null   float64
- 69 Serial_B1050   90 non-null   float64
- 70 Serial_B1051   90 non-null   float64
- 71 Serial_B1054   90 non-null   float64
- 72 Serial_B1056   90 non-null   float64
- 73 Serial_B1058   90 non-null   float64
- 74 Serial_B1059   90 non-null   float64
- 75 Serial_B1060   90 non-null   float64
- 76 Serial_B1062   90 non-null   float64
- 77 GridFins_False   90 non-null   float64
- 78 GridFins_True   90 non-null   float64
- 79 Reused_False   90 non-null   float64
- 80 Reused_True   90 non-null   float64
- 81 Legs_False   90 non-null   float64
- 82 Legs_True   90 non-null   float64
- dtypes: float64(83)
- memory usage: 58.5 KB

SQL Queries

```
--Task 1
--Display the names of the unique launch sites in the space mission
SELECT  DISTINCT(Launch_Site) AS Names
FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1`;

--Task2
--Display 5 records where launch sites begin with the string 'CCA'
SELECT  *
FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1`
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5;

--Task 3
--Display the total payload mass carried by boosters launched by NASA (CRS)
SELECT SUM(PAYLOAD_MASS__KG_)
FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1`
WHERE Payload LIKE '%CRS%'

--Task 4
--Display average payload mass carried by booster version F9 v1.1
SELECT AVG(PAYLOAD_MASS__KG_)
FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1`
WHERE Booster_Version = 'F9 v1.1';
```

SQL Queries

```sql
--Task 5
--List the date when the first succesful landing outcome in ground pad was acheived.
SELECT MIN(Date)
FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1`
WHERE Landing__Outcome = 'Success (ground pad)';

--Task 6
--List the names of the boosters which have success in drone ship and have payload mass
greater than 4000 --but less than 6000

SELECT DISTINCT(Booster_Version)
FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1`
WHERE Landing__Outcome = 'Success (drone ship)' AND  4000 < PAYLOAD_MASS__KG_ AND
PAYLOAD_MASS__KG_ < 6000;
--OR
SELECT DISTINCT(Booster_Version)
FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1`
WHERE Landing__Outcome = 'Success (drone ship)' AND  PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

SQL Queries

```
--Task 7
--List the total number of successful and failure mission outcomes

SELECT
(SELECT COUNT(*) FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1` WHERE Mission_Outcome LIKE
'%Success%') AS success_count,
(SELECT COUNT(*) FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1` WHERE Mission_Outcome LIKE
'%Failure%') AS failure_count
FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1`;

--Task 8
--List the names of the booster_versions which have carried the maximum payload mass. Use a
subquery
SELECT DISTINCT(Booster_Version), PAYLOAD_MASS__KG_
FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1`
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM
`ibmcapstone-376222.SpaceXData.SpaceXTable1`);
--OR
SELECT SUBSTRING( CAST(Date AS STRING), 6, 2) AS month, Landing__Outcome, Booster_Version,
Launch_Site
FROM `ibmcapstone-376222.SpaceXData.SpaceXTable1`
WHERE SUBSTRING(CAST(Date AS STRING), 0, 4)='2015';
```


Thank You!