Falcon 9 Analysis

Dorothy McCord

01-07-2025

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- Methodologies
  - Data Collection with API, Web Scraping
  - EDA with Data Visualization and SQL
  - Interactive Map with Folium
  - Dashboards with Plotly Dash
  - Predictive Analysis
- Results
  - Data Analysis results
  - Maps and Dashboards
  - Predictive results

INTRODUCTION

- Purpose and background

  - The purpose of this presentation is to predict if the first stage Falcon 9 rocket will be successful.

  - SpaceX offers Falcon 9 launches for $62 million, significantly $165 million charged by competitors, largely due to its reusa stage. Predicting the first stage's landing success can help es launch costs, aiding competitors in bidding against SpaceX.

- Challenges

  - From the data, how can successful and failed attempts be de

  - How can the launches be improved to minimize planned and failed launches?

Methodology

Section 2

METHODOLOGY

- Data Collection
  - API data collection
  - Web Scraping info from sources
- Data Wrangling
  - Exploratory analysis
  - Determine training labels
- Data Analysis with visualization and SQL
- Interactive analytics with Folium and Plotly Dash
- Predictive analysis
  - Processing data to make predictions

Data Collection

#	Step	Code
1.	Request response from API	spacex_url="https://api.spacexdata.om/v4/launches/past"
2.	Convert data to JSON file	response=requests.get(static_json_url).json(
		data=pd.json_normalize(response)
3.	Transform data	getBoosterVersion(data)
		getLaunchSite(data)
		getPayloadData(data)
		getCoreData(data)
4.	Create dictonary	launch_dict = {'FlightNumber': list(data['flight_number']),
		'Date': list(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}
5.	Create dataframe	df = pd.DataFrame.from_dict(launch_dict)
6.	Filter dataframe	data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
7.	Export file	data_falcon9.to_csv('dataset_part_1.csv', index=False)

Data Scraping

1.	Response from HTML	data = requests.get(static_url).text
2.	Create BeautifulSoup Object	soup = BeautifulSoup(data, 'lxml')
3.	Find tables	html_tables = soup.find_all('table')
4.	Get 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)
5.	Create dictionary	launch_dict= dict.fromkeys(column_names)
6.	Add data to keys	launch_dict= dict.fromkeys(column_names)
7.	Create dataframe	df=pd.DataFrame(launch_dict)
8.	Export file	df.to_csv('spacex_web_scraped.csv', index=F

Data Wrangling

1.	Request response from API	spacex_url="https://api.spacexdata.om/v4/launches/past"
2.	Convert data to JSON file	response=requests.get(static_json_url).json(
		data=pd.json_normalize(response)
3.	Transform data	getBoosterVersion(data)
		getLaunchSite(data)
		getPayloadData(data)
		getCoreData(data)
4.	Create dictonary	launch_dict = {'FlightNumber': list(data['flight_number']),
		Date': list(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}
5.	Create dataframe	df = pd.DataFrame.from_dict(launch_dict)
6.	Filter dataframe	data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
7.	Export file	data_falcon9.to_csv('dataset_part_1.csv', index=False)

EDA with Data Visualization

Bar Graph
numeric and categoric variables.
Success rate vs orbit
Bar graphs show relationships between

Line Graph
Success rate vs year
Line graphs show data variables and their trends.

Scatter Graphs
Flight number vs payload mass
Flight number vs launch site
Payload vs launch site
Payload vs orbit type
Orbit vs flight number
Orbit vs payload mass
Scatter plots show relationships betw

EDA with SQL

- SQL queries were performed to gather and understand data from the da[ta]

  - Displaying the names of the unique launch sites in the space mission.
  - Display 5 records where launch sited 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 achi[eved]
  - List the names of the boosters which have success in drone ship and have paylo[ad] than 4,000 but less than 6,000.
  - List the total number of successful and failure missions.
  - List the names of the booster versions which have carried the maximum payloa[d]
  - List the records which will display the month names, failure landing, outcomes booster versions, launch site for the months in year 2015.
  - Rank the count of successful landing outcomes between the date June 4, 2010 in descending order.

Interactive Map with Folium
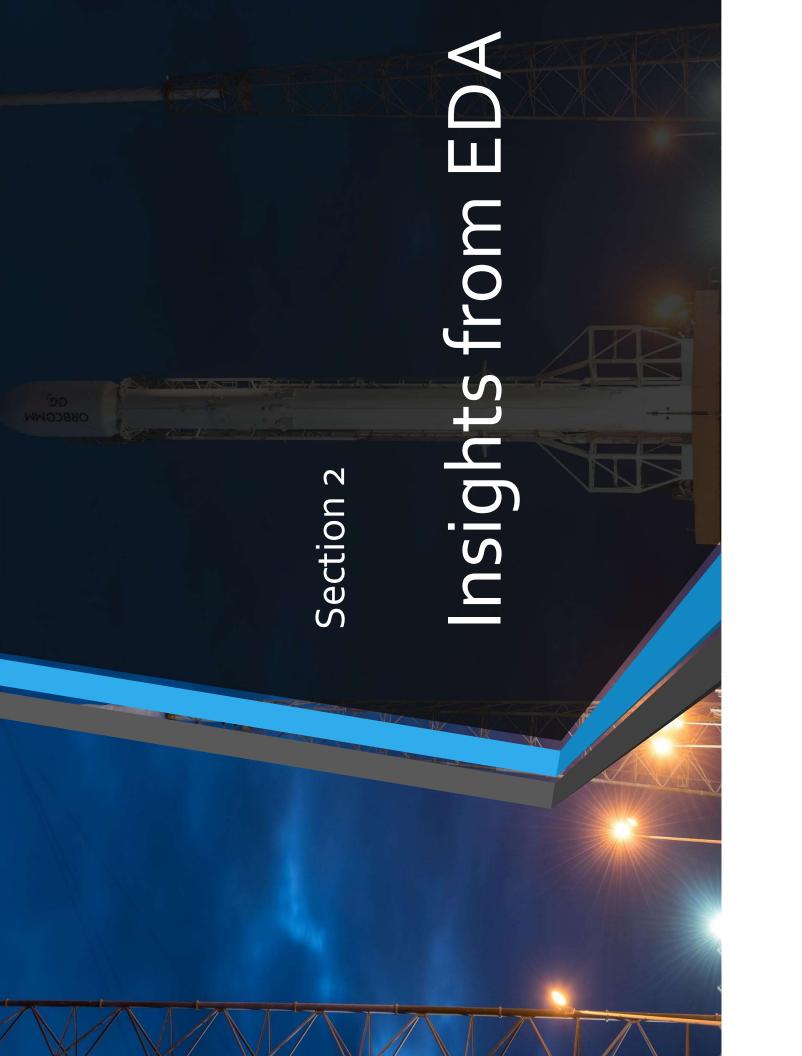
- Folium map object is a map centered on NASA Johnson Space Cen... Houston, Texas

  ○ Purple circle at NASA Johnson Space Center's coordinate with label showin...

  ○ Purple circles at each launch site coordinates with label showing launch sit...

  ○ The grouping of points in a cluster to display multiple and different informa... same coordinates.

  ○ Markers to show successful and unsuccessful landings. Green for successfu... and Red for unsuccessful landing.

  ○ Markers to show distance between launch site to key locations and plot a li... them.

Dashboard with Plotly Dash
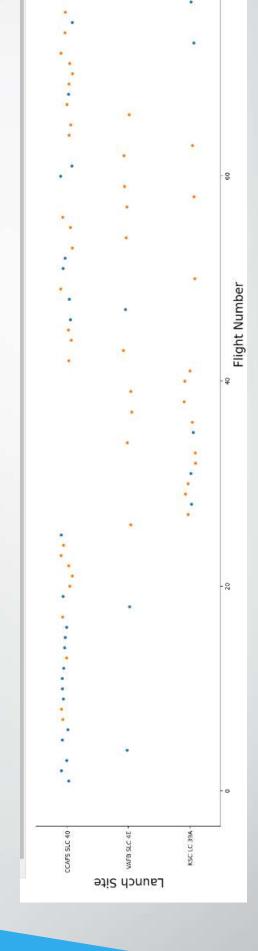
- Dashboard has dropdown, pie chart, rangeslider and scatter plot components

  ○ Dropdown allows user to choose the launch site or all launch sites.

  ○ Pie chart shows the total success and the total failure for the launch site ch the dropdown component.

  ○ Rangeslider allows a user to select a payload mass in a fixed range.

  ○ Scatter chart shows the relationship between two variables, in particular S Payload Mass.

Predictive Analysis
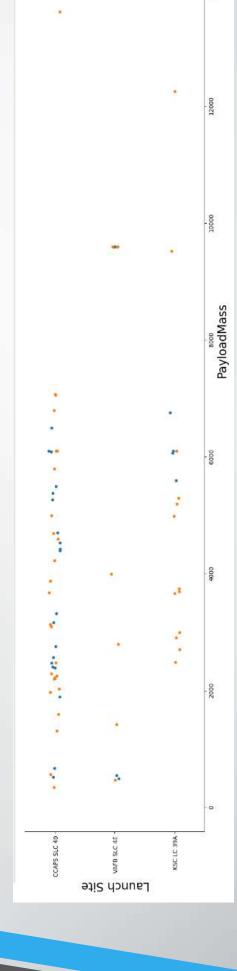
- Data preparation
  - Load datasheet
  - Normalize data
  - Split data into training and test sets
- Model preparation
  - Selection of machine learning algorithms
  - Set parameters for each algorithm to GridSearchCV
  - Training GridSearchModel models with training dataset
- Model evaluation
  - Get best hyperparameters for each type of model
  - Compute accuracy for each model with test dataset
  - Plot confusion matrix
- Model comparison
  - Comparison of models according to their accuracy
  - The model with the best accuracy will be chosen.

Section 2

Insights from EDA

Flight Number vs Launch Site
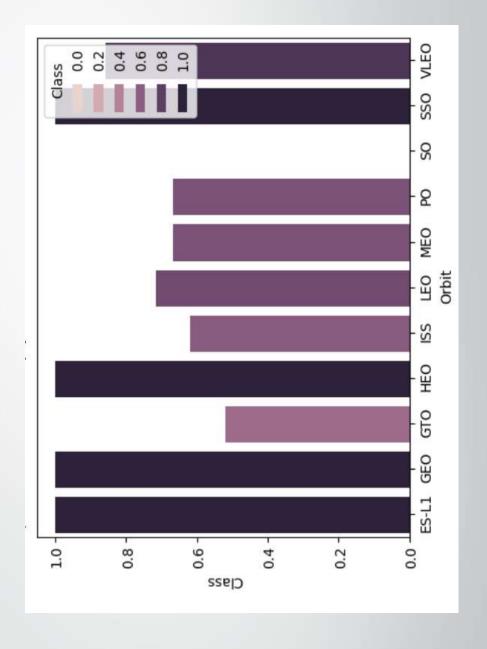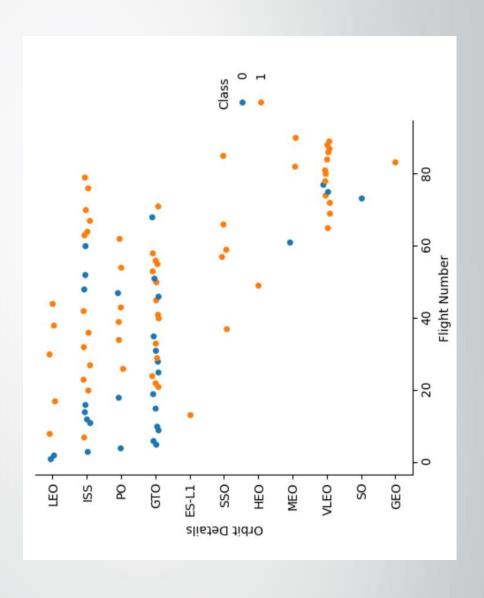
Payload vs Launch Site
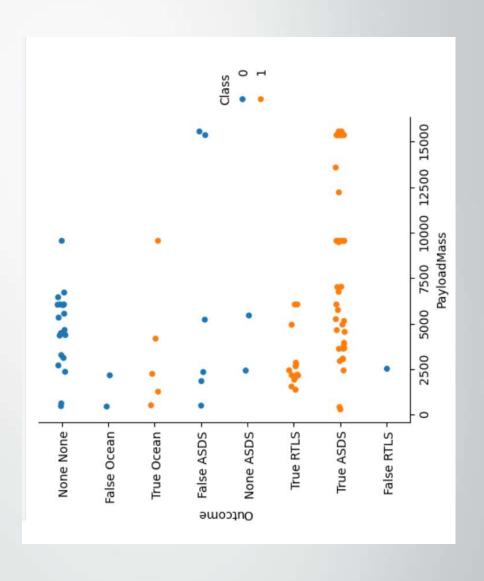
Success Rate vs Orbit Type
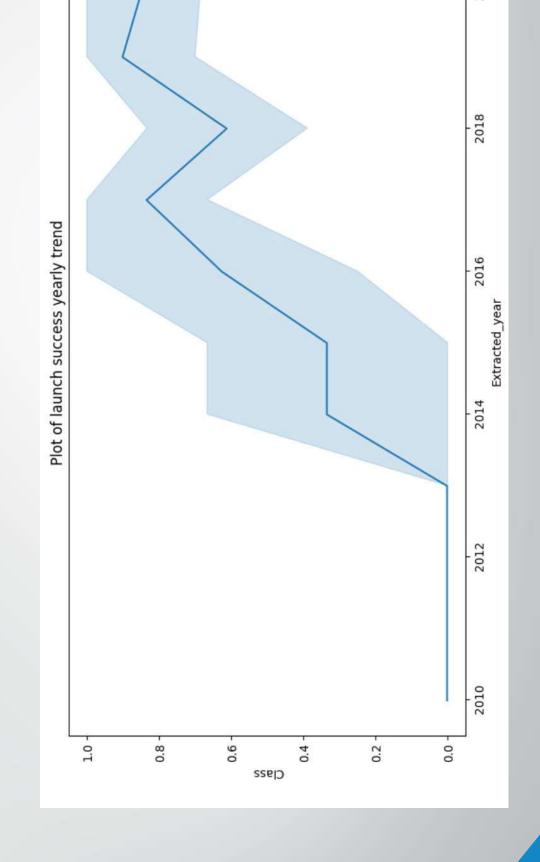
Flight Number vs Orbit Type

Payload vs Orbit Type

Launch Success Yearly Trend

Plot of launch success yearly trend

Launch site names

```
%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

Launch Site Names Beginning with "C

```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

* sqlite:///my_data1.db
Done.

Launch_Site

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;

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

payloadmass

619967

Average Payload Mass – F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

* sqlite:///my_data1.db
Done.

payloadmass
6138.28712871287

First Successful Ground Landing Date

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

* sqlite:///my_data1.db
Done.

min(DATE)
2010-06-04

Successful Drone Ship Landing with Payload between 4,o

```
# checking column names
%sql PRAGMA table_info(SPACEXTBL);

# Commented out code to check column names first
%sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_
```

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

Booster_Version

F9 FT B1022

F9 FT B1026

F9 FT B1021.2

F9 FT B1031.2

Total Number of Successful and Failure Mission O

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

* sqlite:///my_data1.db
Done.

missionoutcomes

missionoutcomes
1
98
1
1

Booster Carried Maximum Payload

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

* sqlite:///my_data1.db
Done.

boosterversion

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

2015 Launch Records

```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where EXTRACT(YEAR F

 * sqlite:///my_data1.db
(sqlite3.OperationalError) near "FROM": syntax error
[SQL: SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where EXTRACT(YEAR FR
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

Rank Landing Outcomes Between 4/6/2010 and

```
%sql SELECT Landing_Outcome FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

* sqlite:///my_data1.db
Done.

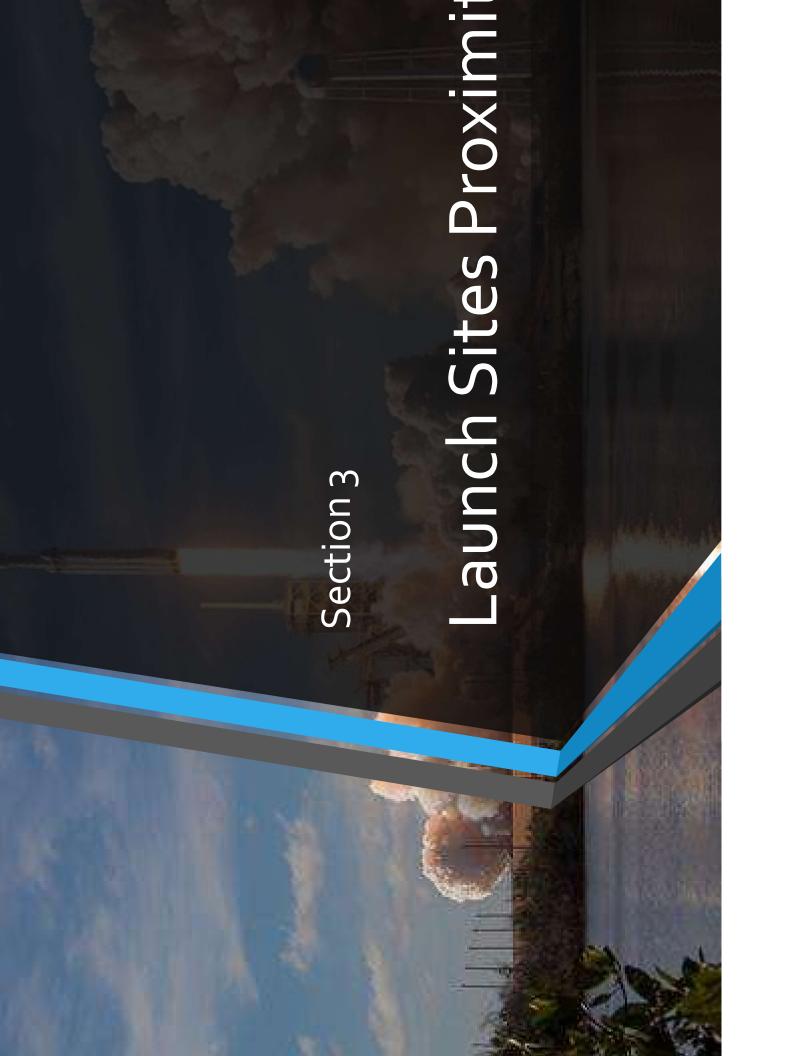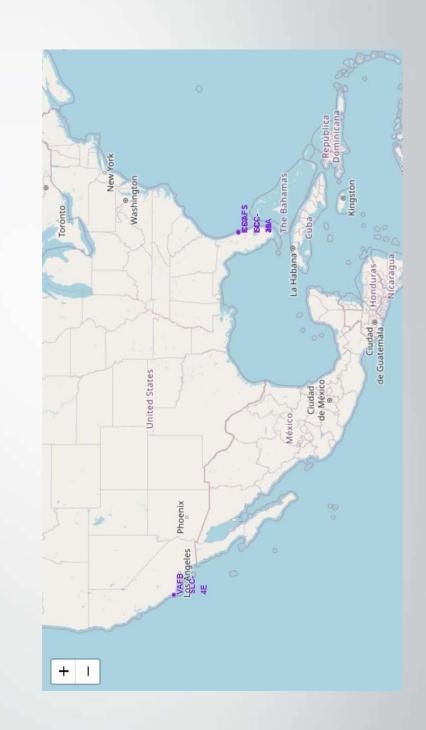| Landing_Outcome |
|---|
| No attempt |
| Success (ground pad) |
| Success (drone ship) |
| Success (drone ship) |
| Success (ground pad) |
| Failure (drone ship) |
| Success (drone ship) |
| Success (drone ship) |
| Success (drone ship) |
| Failure (drone ship) |
| Failure (drone ship) |
| Success (ground pad) |
| Precluded (drone ship) |
| No attempt |
| Failure (drone ship) |
| No attempt |
| Controlled (ocean) |

Launch Sites Proximit
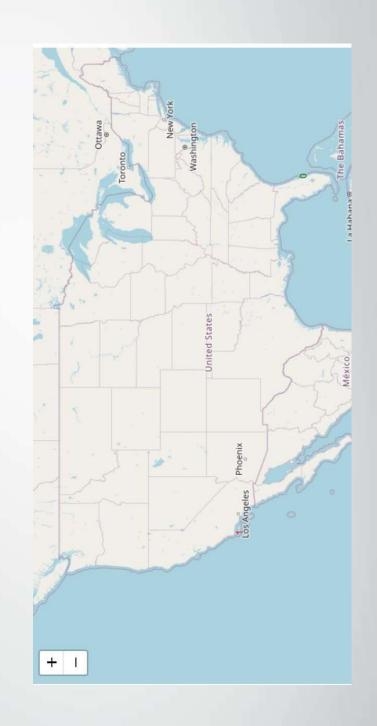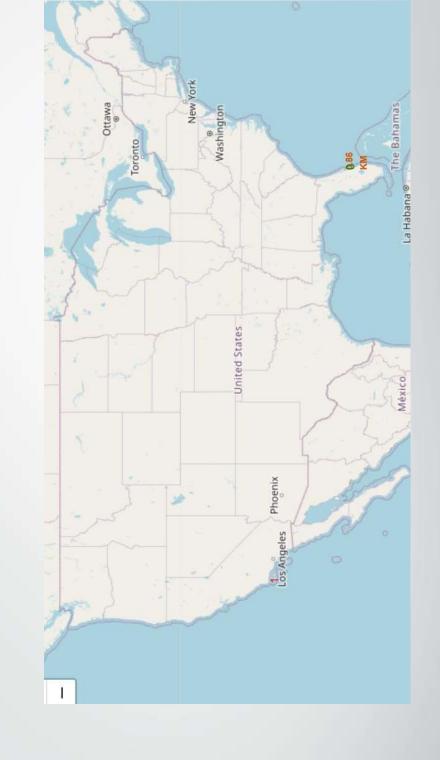
Folium map – Ground Stations

Folium map – Color Labeled Marker

Folium map – Distances between CCAFS SLC-40 and its p
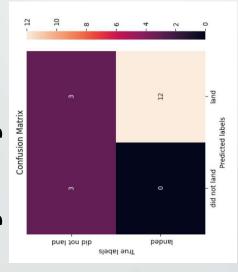
Predictive Analysis

Section 4

Testing Accuracy
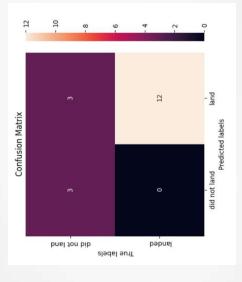
```python
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}

parameters ={"C":[0.01,0.1,1], 'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()

logreg_cv = GridSearchCV(lr,parameters,cv=10)
logreg_cv.fit(X_train, Y_train)

GridSearchCV(cv=10, estimator=LogisticRegression(),
             param_grid={'C': [0.01, 0.1, 1], 'penalty': ['l2'],
                         'solver': ['lbfgs']})
```

Confusion Matrix

Decision Tree

SVM

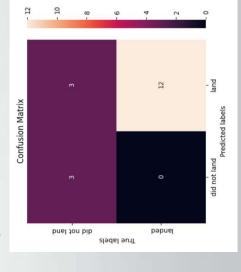## Logistic Regression
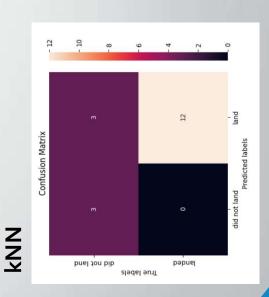

kNN

Conclusions

- The success of a mission can be explained by several factors.

- The orbits with the best success rates are GEO, HEO, SSO, E5-L1.

- The payload mass can be a criterion to take into account for the su... mission. Low weighted payloads perform better than the heavy w... payloads.

- With the current data, we cannot explain why some launch sites a... than others. Atmospheric data would increase predictability.

- The Decision Tree Algorithm was used for this dataset because it h... better train accuracy.