Data Science Capstone Project

Daksh Vashist

https://www.github.com/dakshvashist

12/01/2024

Outline

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

Executive Summary

- Data collection from SpaceX API and Wikipedia page.
- Data exploration using SQL, utilized Dash and Folium for data visualization. Gathered relevant columns to be used as features.
- One hot encoding.
- Used GridSearchCV for optimal hyperparameters.

- ML models used: Logistic Regression, SVM, KNN, and Decision Tree Classifier. All the models produced similar results with accuracy rate of about 83.33%. More data is needed for better model determination and accuracy. Type 1 problem persistent for models.

Introduction

- Space X has best pricing ($62 million vs. $165 million USD)
- The ability to recover part of rocket (Stage 1)
- Space Y wants to compete with Space X

Problem:

- We help Space Y train a machine learning model to predict the successful Stage 1 landing

Methodology

- Data collection methodology:
  - Collected data from SpaceX API and Wikipedia page
- Data wrangling
- Exploratory data analysis using SQL and Python
- Data visualization using Plotly Dash and Folium
- Perform predictive analysis using classification models

Methodology

Overview of data collection, wrangling, visualization, dashboard, and model methods

Data Collection – SpaceX API

1. Request (SpaceX API)
2. Obtain .JSON file
3. Convert json file to pandas DataFrame
4. Filter data
5. Impute missing values

Data Collection – Web Scraping

1. Request HTML data
2. Parse data using Beautiful Soup
3. Iterate data to extract into a dictionary
4. Cast the dictionary into a DataFrame

EDA with Data Visualization

Exploratory Data Analysis performed on various variables

Plots Used:

Scatter plots, line charts, and bar plots used to visualize relationships between variables

Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit vs. Success Rate, Flight Number vs. Orbit, Payload vs Orbit, and Success Yearly Trend

EDA with SQL

- Imported into IBM Database.
- Data queries using SQL Python(SQLAlchemy)
- Queries helped attain a finder understanding of the dataset
- Queried information about launch site names, various pay load sizes of customers and booster versions, mission outcomes, and landing outcomes

Build an interactive map with Folium

Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.

This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.

Build a Dashboard with Plotly Dash

Dashboard includes a pie chart and a scatter plot.

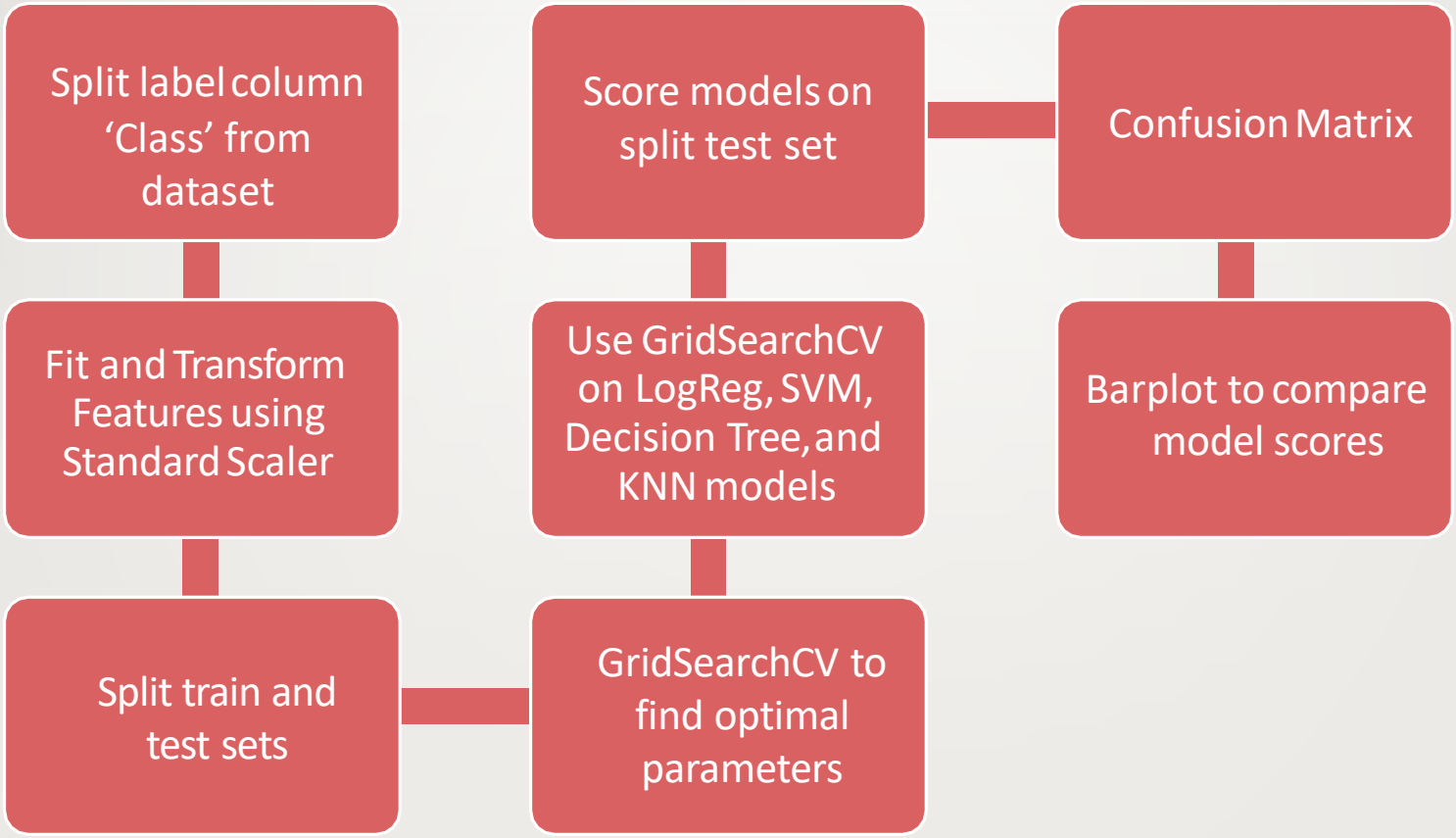
Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.

Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.

The pie chart is used to visualize launch site success rate.

The scatter plot can help us see how success varies across launch sites, payload mass, and
booster version category.

Predictive analysis (Classification)

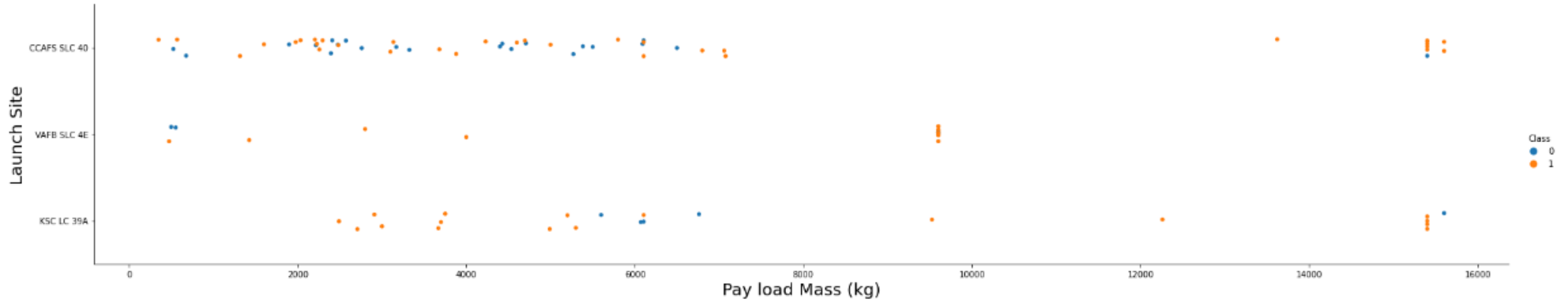


EDA with Visualization

Exploratory data analysis with seaborn plots

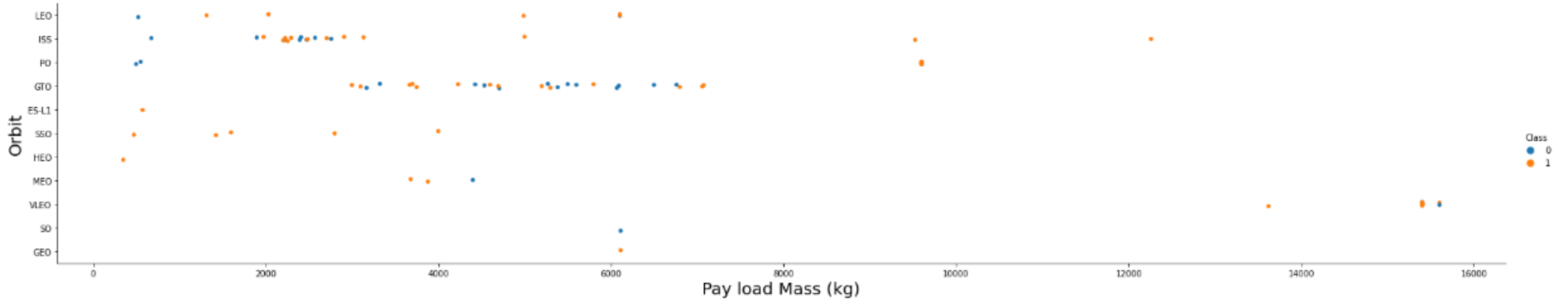
Payload vs. Launch Site

In [ ]:
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(x="PayloadMass", y="LaunchSite", hue="Class", data=df, aspect = 5)
plt.ylabel("Launch Site",fontsize=20)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



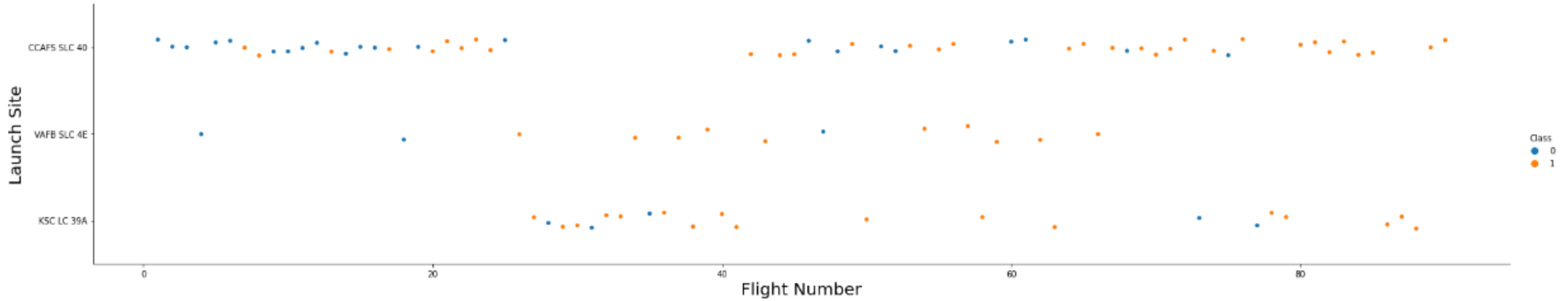
Payload vs. Orbit type

In [ ]:
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="PayloadMass", y="Orbit", hue="Class", data=df, aspect = 5)
plt.ylabel("Orbit",fontsize=20)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



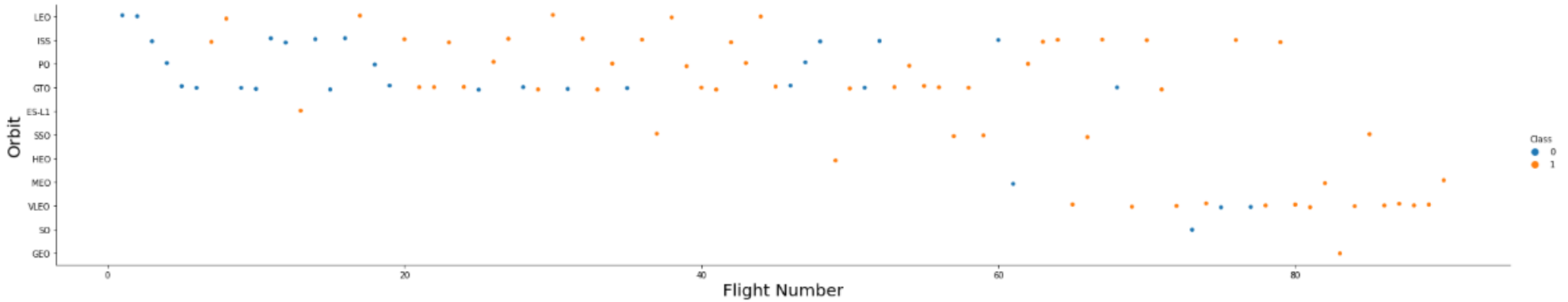
Flight Number vs. Launch Site

In [ ]:
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



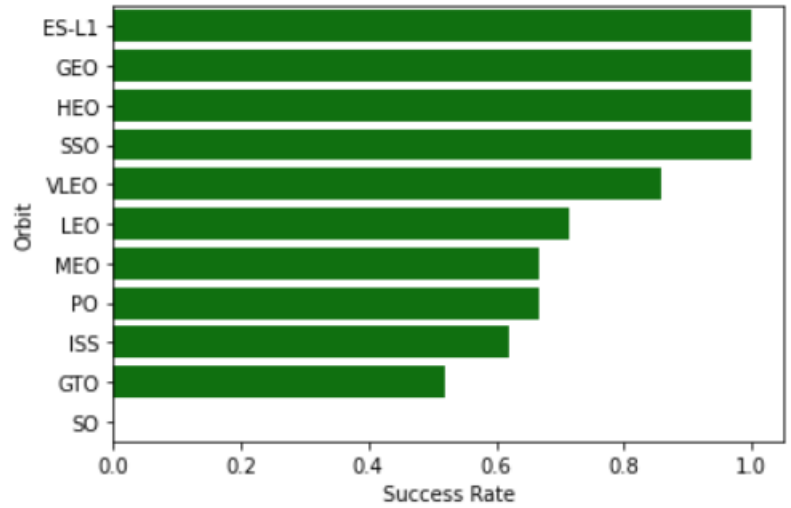
Flight Number vs. Orbit type

In [ ]:
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="FlightNumber", y="Orbit", hue="Class", data=df, aspect = 5)
plt.ylabel("Orbit",fontsize=20)
plt.xlabel("Flight Number",fontsize=20)
plt.show()
```

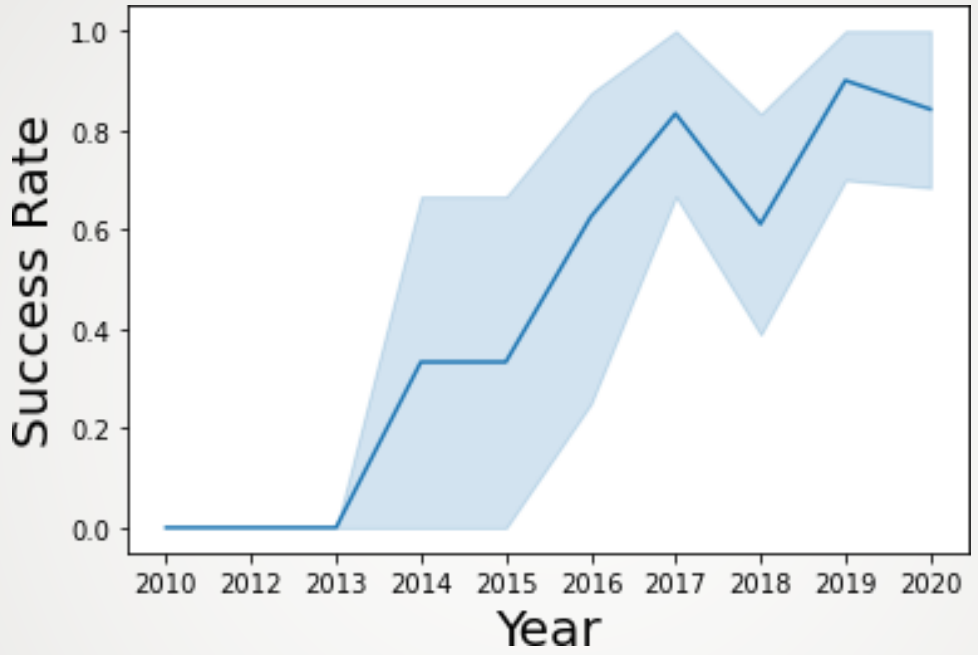


Success rate vs. Orbit type

In [ ]:
```
# HINT use groupby method on Orbit column and get the mean of Class column
success_by_orbit = df[['Orbit', 'Class']].groupby('Orbit').mean()
success_by_orbit.sort_values(by='Class', inplace=True, ascending=False)#.plot(kind='barh', color='g')
success_by_orbit.reset_index(inplace=True)
success_by_orbit
sns.barplot(x='Class', y='Orbit', data=success_by_orbit, color='g')
plt.xlabel('Success Rate')
plt.show()
```



Launch Success Yearly Trend



EDA with SQL

Exploratory data analysis in python with SQLAlchemy

All Launch Site Names

Display the names of the unique launch sites in the space mission

In [18]:
```
%%sql
Select distinct Launch_Site
From SPACEXTBL
```

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

Out[18]:

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

Query unique launch site names from database.

Launch Site Names Beginning with `CCA`

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

In [21]:
```
%%sql
Select *
From SPACEXTBL
Where Launch_Site like 'CCA%'
Limit 5
```

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

Out[21]:

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

Total Payload Mass from NASA

In [24]:
```
%%sql
Select sum(PAYLOAD_MASS__KG_) as Total_pyld_mass
From SPACEXTBL
Where Customer = 'NASA (CRS)'
```

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

Out[24]:

Total_pyld_mass
45596

Average Payload Mass by Falcon 9

```
In [26]: %%sql
         Select avg(PAYLOAD_MASS__KG_) as Avg_pyld_mass
         From SPACEXTBL
         Where Booster_Version like 'F9 v1.1%'

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

Out[26]:

Avg_pyld_mass
2534.6666666666665

This query calculates the average payload mass or launches which used booster version Falcon 9

First Successful Ground Pad Landing Date

In [35]:
```
%%sql
Select min(Date) as Fst_Successful_landing
From SPACEXTBL
Where "Landing _Outcome" = 'Success (ground pad)'
```

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

Out[35]:

Fst_Successful_landing
01-05-2017

Successful drone ship landing with payload between 4000 and 6000

In [37]:
```
%%sql
Select Booster_Version
From SPACEXTBL
Where "Landing _Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ Between 4000 and 6000
```

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

Out[37]:

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

Total Number of Each Mission Outcome

In [58]:
```
%%sql
Select Mission_Outcome, count(Mission_Outcome) as Outcome_count
From SPACEXTBL
Group by Mission_Outcome
Order by 2 desc
```

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

Out[58]:

Mission_Outcome	Outcome_count
Success	99
Success (payload status unclear)	1
Failure (in flight)	1

Boosters that carried Maximum Payload

In [42]:
```
%%sql
Select Booster_Version
From SPACEXTBL
Where PAYLOAD_MASS__KG_ = (
    Select Max(PAYLOAD_MASS__KG_)
    From SPACEXTBL
)
```

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

Out[42]:

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

2015 Failed Drone Ship Landing Records

In [44]:
```
%%sql
Select substr(Date, 4, 2) as month, "Landing _Outcome", Booster_Version, Launch_Site
From SPACEXTBL
Where substr(Date,7,4)='2015' and "Landing _Outcome" = 'Failure (drone ship)'
```

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

Out[44]:

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

Ranking Counts of Successful Landings Between 2010-06-04 and 2017-03-20

In [59]:
```
%%sql
Select
    "Landing _Outcome", count("Landing _Outcome") as cnt
From SPACEXTBL
Where Date Between '04-06-2010' and '20-03-2017'
Group by "Landing _Outcome"
Order by 2 desc
```

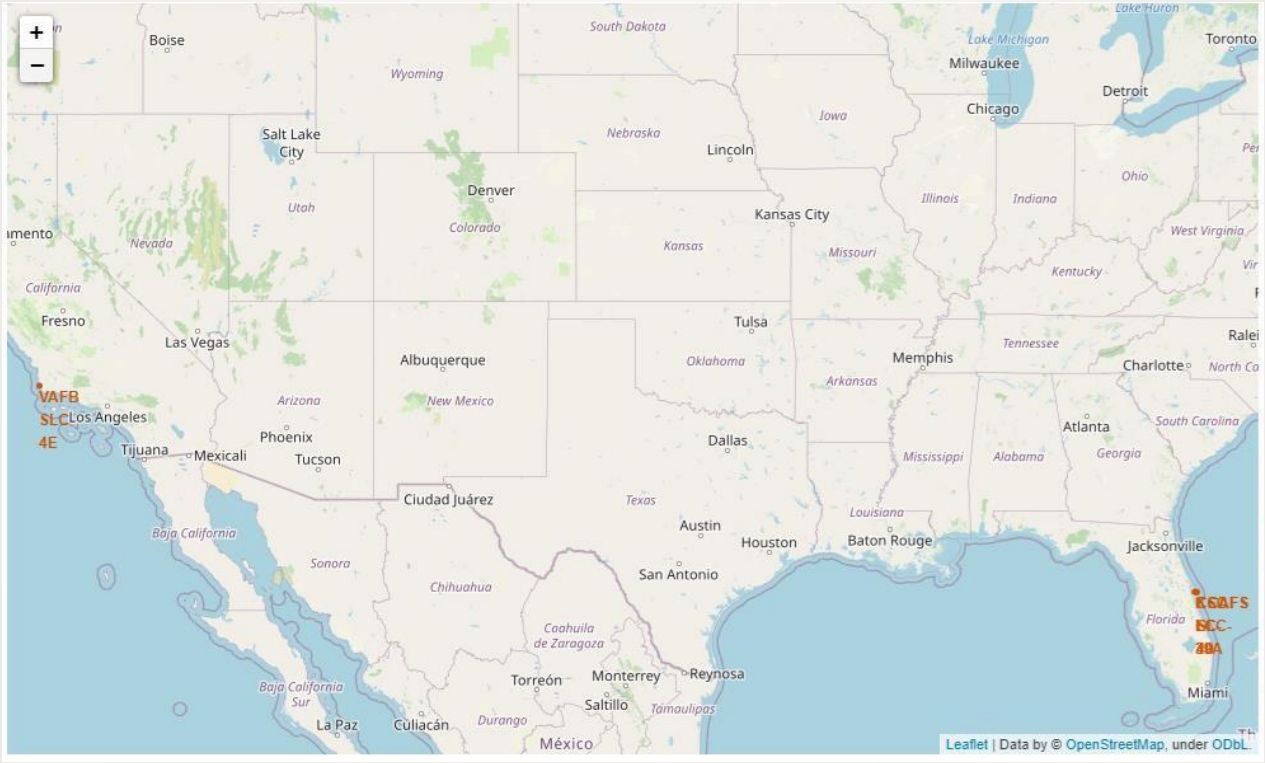
```
 * sqlite:///my_data1.db
Done.
```

Out[59]:

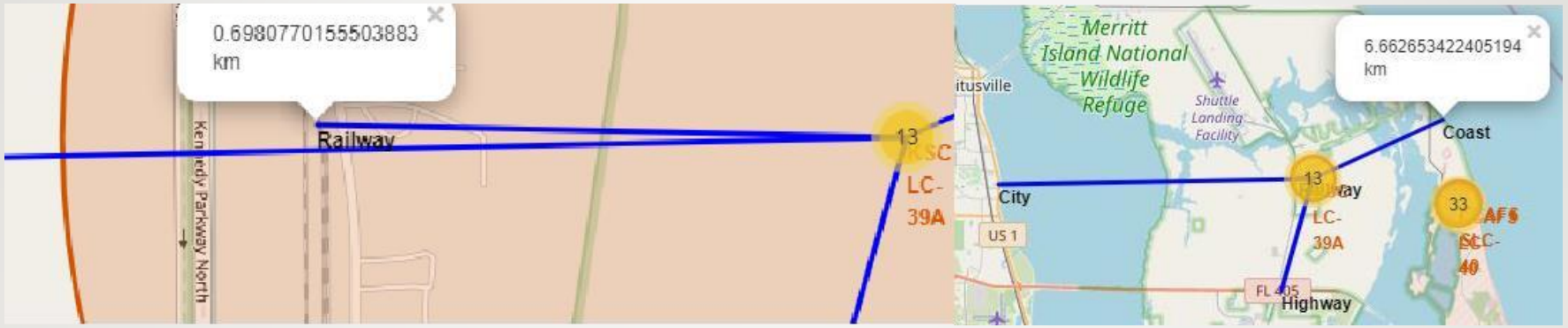
Landing _Outcome	cnt
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

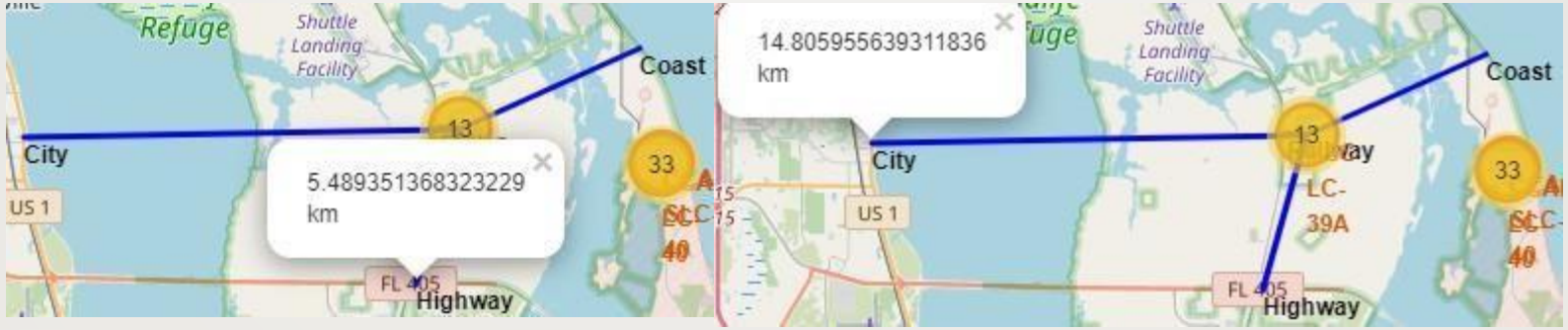
Interactive Map with Folium

Launch Site Location

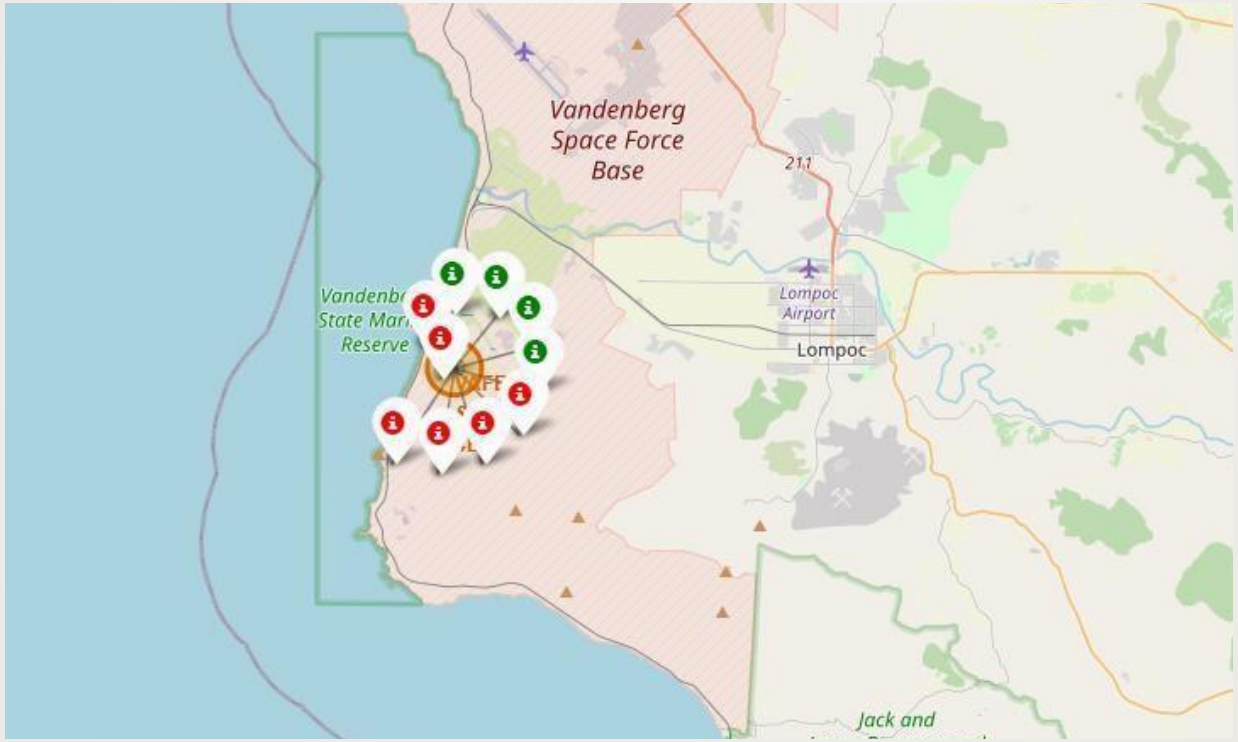


Key Location Proximities



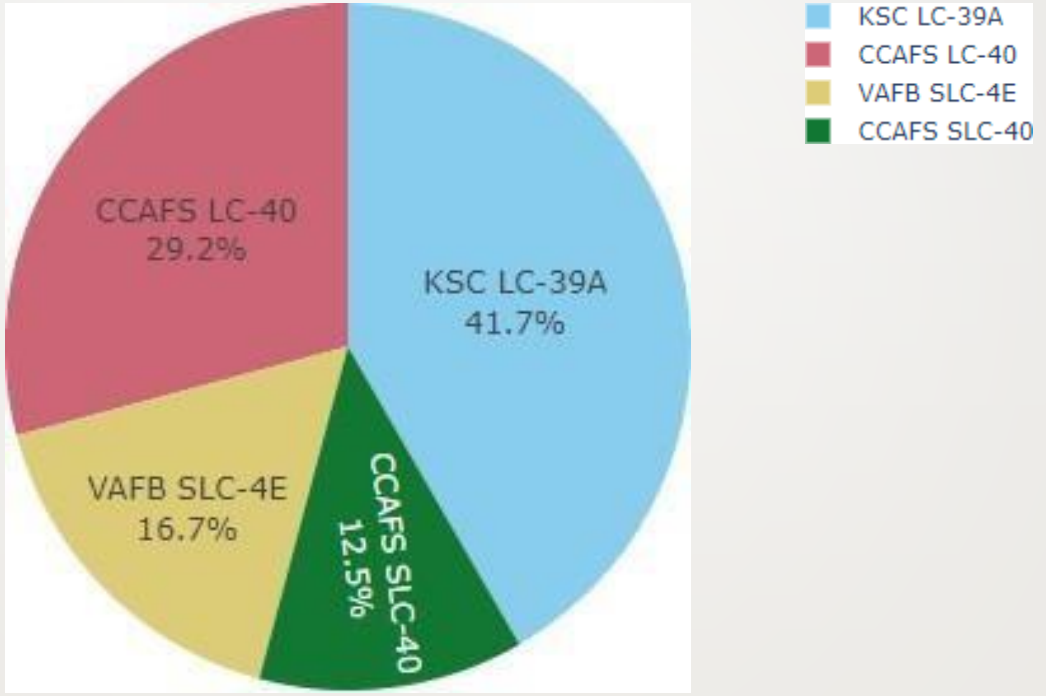


Color-Coded Launch Markers

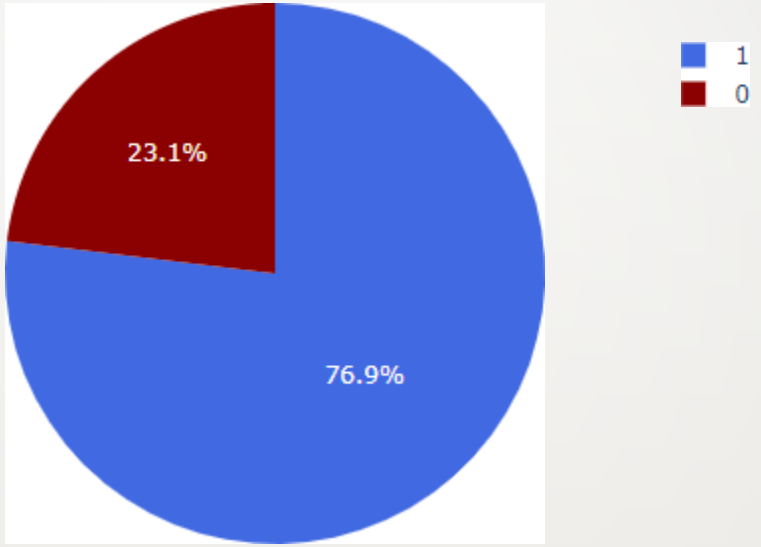


Building a Dashboard with Plotly Dash

Successful Launches Across Launch Sites



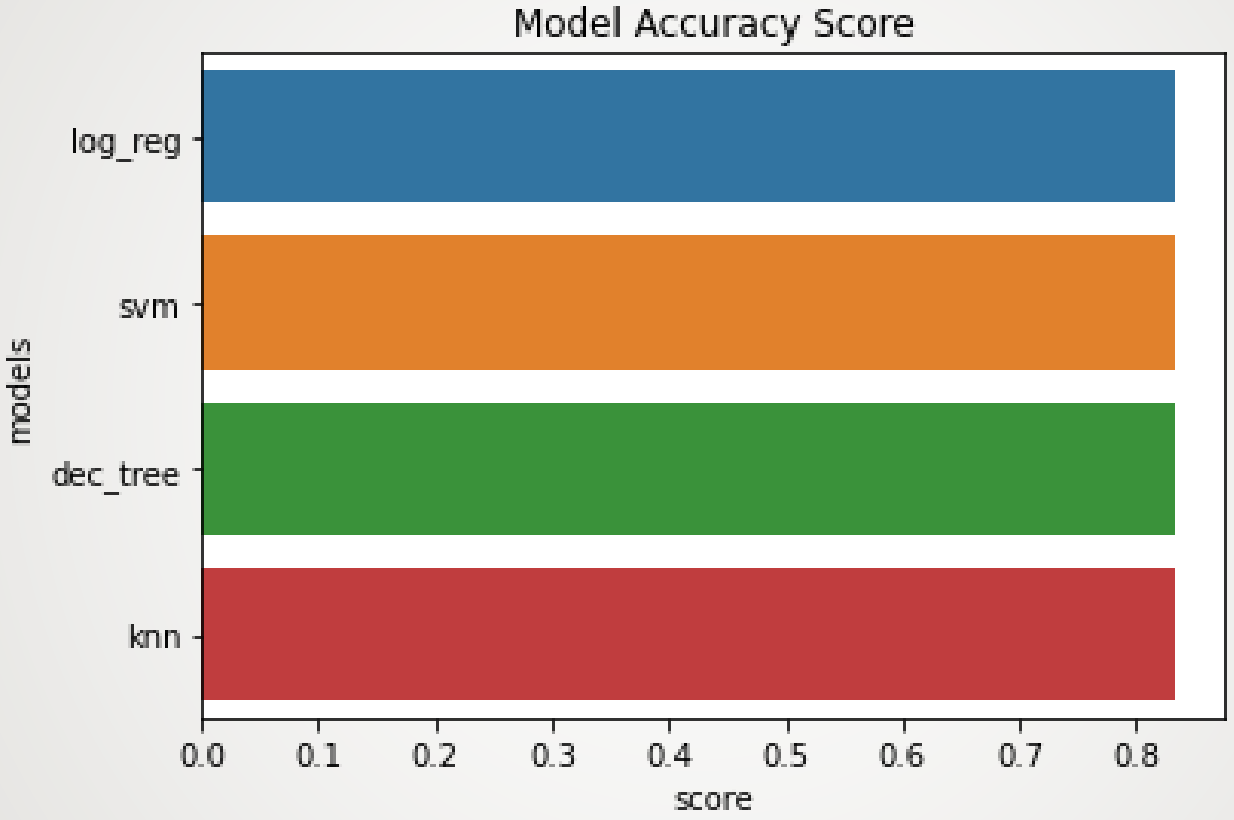
Highest Success Rate Launch Site



KSC LC-39A has the highest success rate with 10 successful landings (blue) and 3 failed landings (red).

- Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix

All 4 models performed similar for the test set, the confusion matrix is the same across all models.

APPENDIX

GitHub: https://github.com/dakshvashist/IBM-Data-Science-Professional-Certificate

Instructors:

Instructors: Rav Ahuja, Alex Aklson, Aije Egwaikhide, Svetlana Levitan, Romeo Kienzler, Polong Lin, Joseph Santarcangelo, Azim Hirjani, Hima Vasudevan, Saishruthi Swaminathan, Saeed Aghabozorgi, Yan Luo