IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sreetama C.

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection via web scraping, SpaceX API
  - Data wrangling
  - Exploratory data analysis via SQL, visualization
  - Interactive visual analytics via Folium
  - Machine learning
- Summary of all results
  - Screenshots of EDA, visual analytics, ML predictions

Introduction

- Investigation into SpaceX rocket launch methodology and its effectiveness
  - SpaceX reuses the “first stage” of a rocket in future launches, which increases efficiency and decreases cost
- Rival company SpaceY attempting to compete — ML pipeline to predict whether or not the reuptake of a rocket’s first stage is a viable landing outcome
  - Highly contributes to ability to compete with or match SpaceX
- Identifying factors that affect rocket launch outcome, how each factor affects the outcome, and what best conditions for a successful landing are
- All notebooks can be found here

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API and Wikipedia web scraping
- Perform data wrangling
  - One-hot encoding of categorical variables
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium
- Perform predictive analysis using classification models

Data Collection – SpaceX API

SpaceX REST API GET request to obtain data and normalize into Pandas dataframe

Filter dataframe to only include Falcon 9 launches

Clean data and reformat into neater dataframe

Data Collection - Scraping

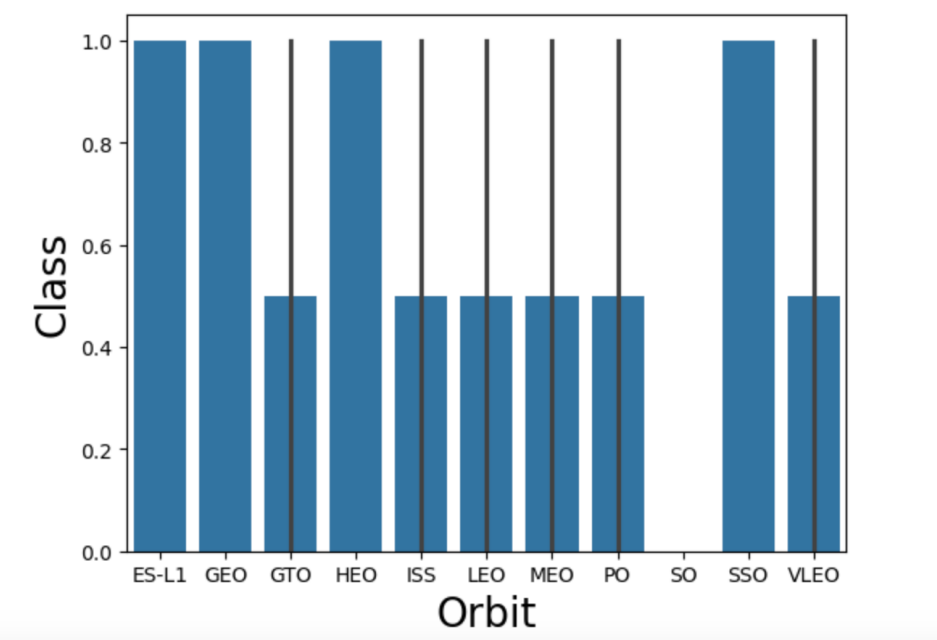
Request Falcon 9 Wikipedia page via URL and HTTP GET method

Use BeautifulSoup and helper functions to reformat into readable HTML

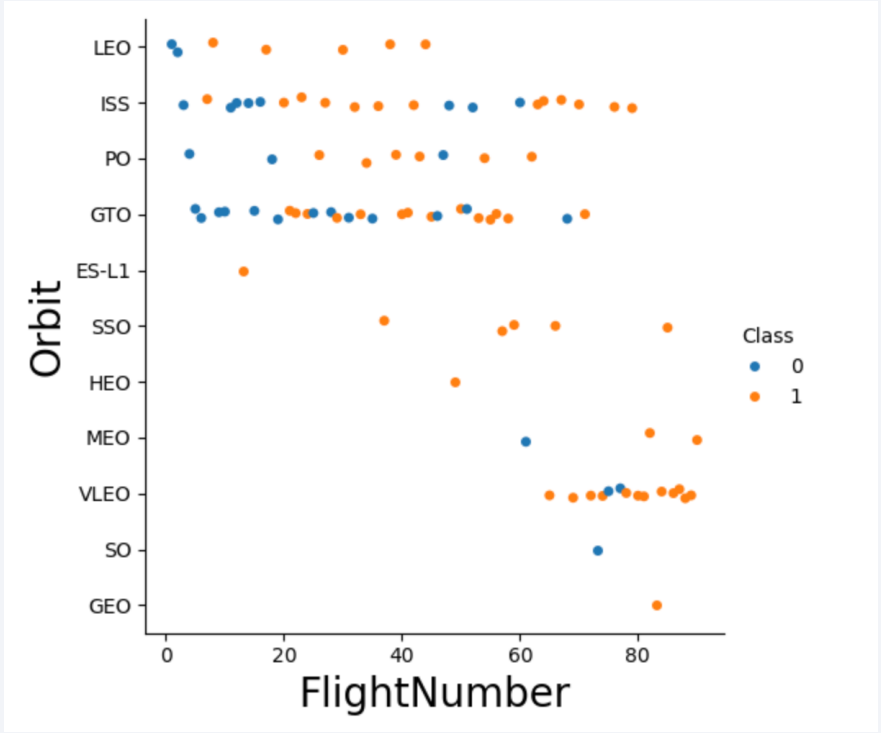
Parse data into a dictionary and convert to a Pandas dataframe

EDA with Data Visualization

- Visualizations of relationships between various features
  - Flight number vs orbit type, orbit type vs class, success over time, payload mass vs orbit time, etc.
  - Bar plots for categorical variables, line graphs for change over time, scatterplots to visualize clusters of data points
- Feature engineering: one-hot encoding of categorical variables



Pictured: bar graph of orbit type in relation to class (left), scatter plot of flight number in relation to orbit type (right)



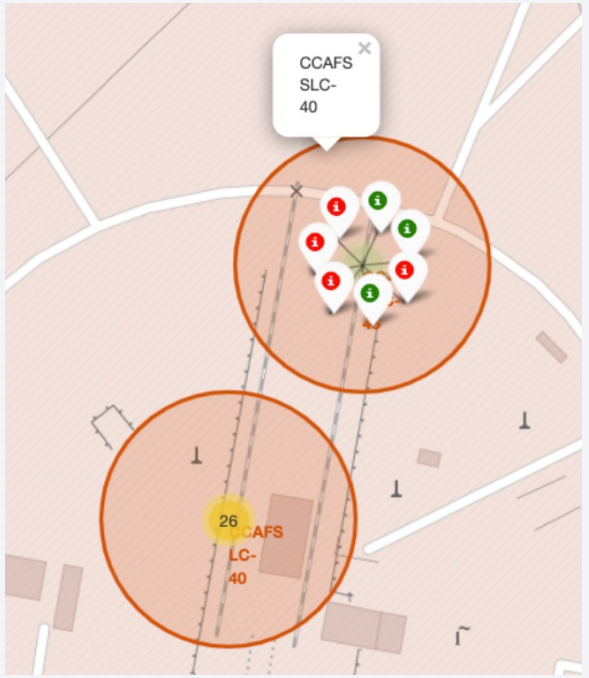
EDA with SQL

- Among the SQL queries performed:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites began with the string 'CCA'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the names of the boosters which have succeeded in drone ship and have a payload mass between 4000 and 6000
  - Listing the names of the booster_versions which have carried the maximum payload mass using a subquery
  - Ranking 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

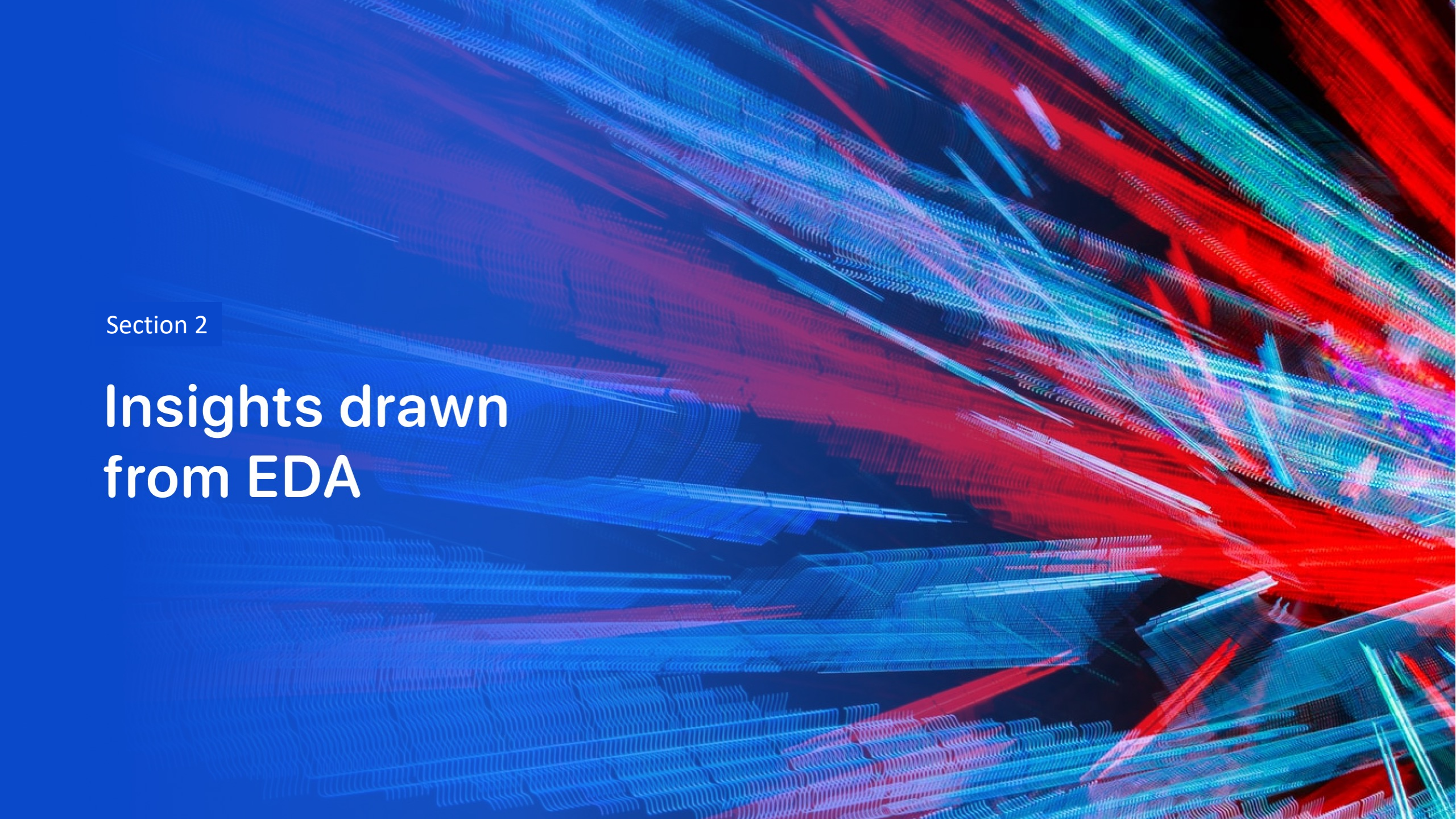
Build an Interactive Map with Folium

- Previously used: matplotlib, seaborn for visual exploration
- Folium maps for interactivity and geographic observations
- Mapped and marked launch sites, failed & successful launches for each launch site, distance from each launch site to its proximities
- Launch sites often tend to be coastal and close to the Equator

Pictured: map of launch sites (left) and zoomed-in success-failure markers (right)

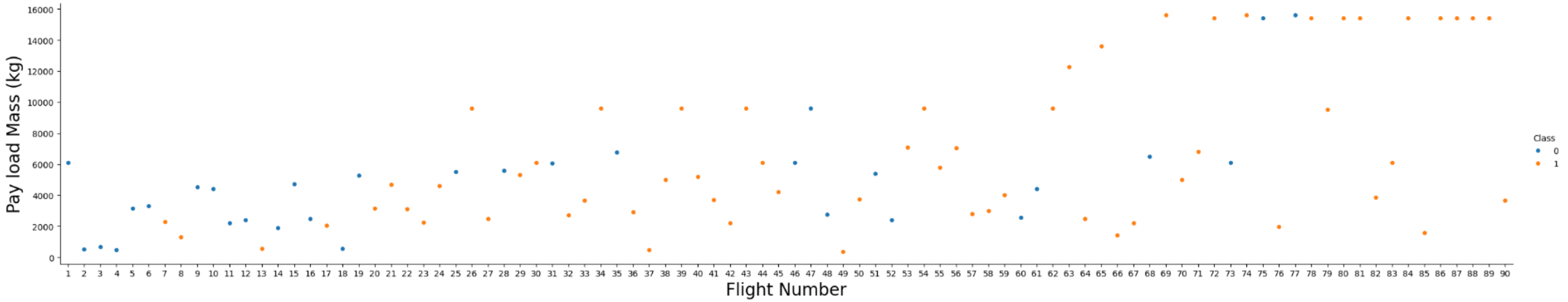


Predictive Analysis (Classification)


Section 2
Insights drawn from EDA

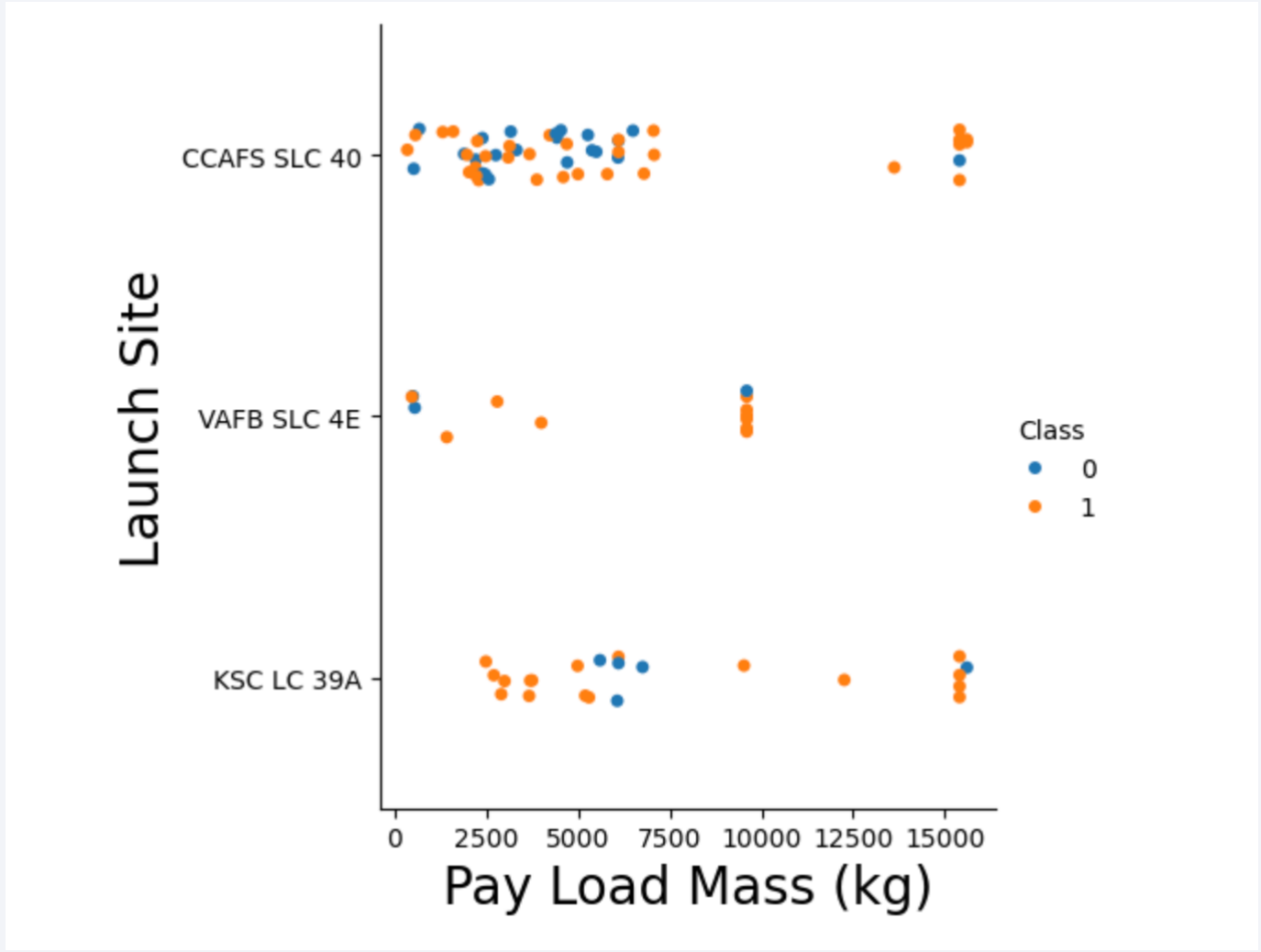
Flight Number vs. Launch Site

```
sns.catplot(y="PayloadMass", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



Payload vs. Launch Site

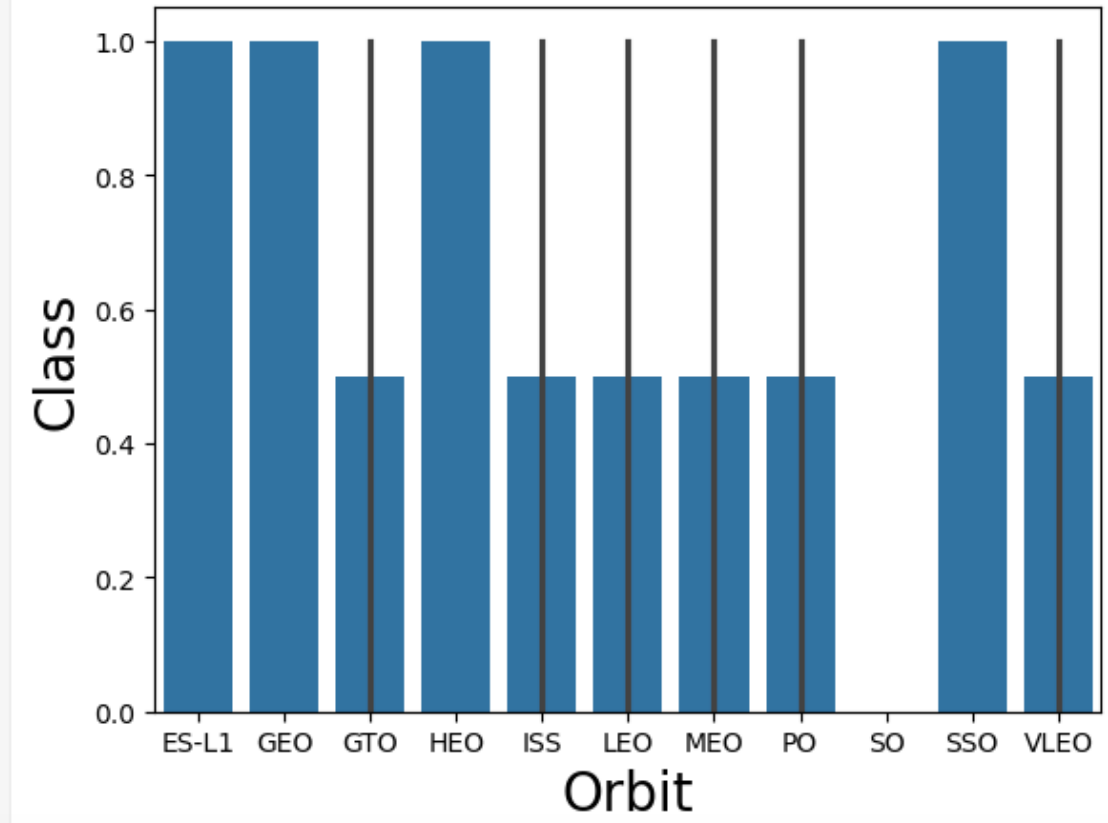
```
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df)
plt.xlabel("Pay Load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Success Rate vs. Orbit Type

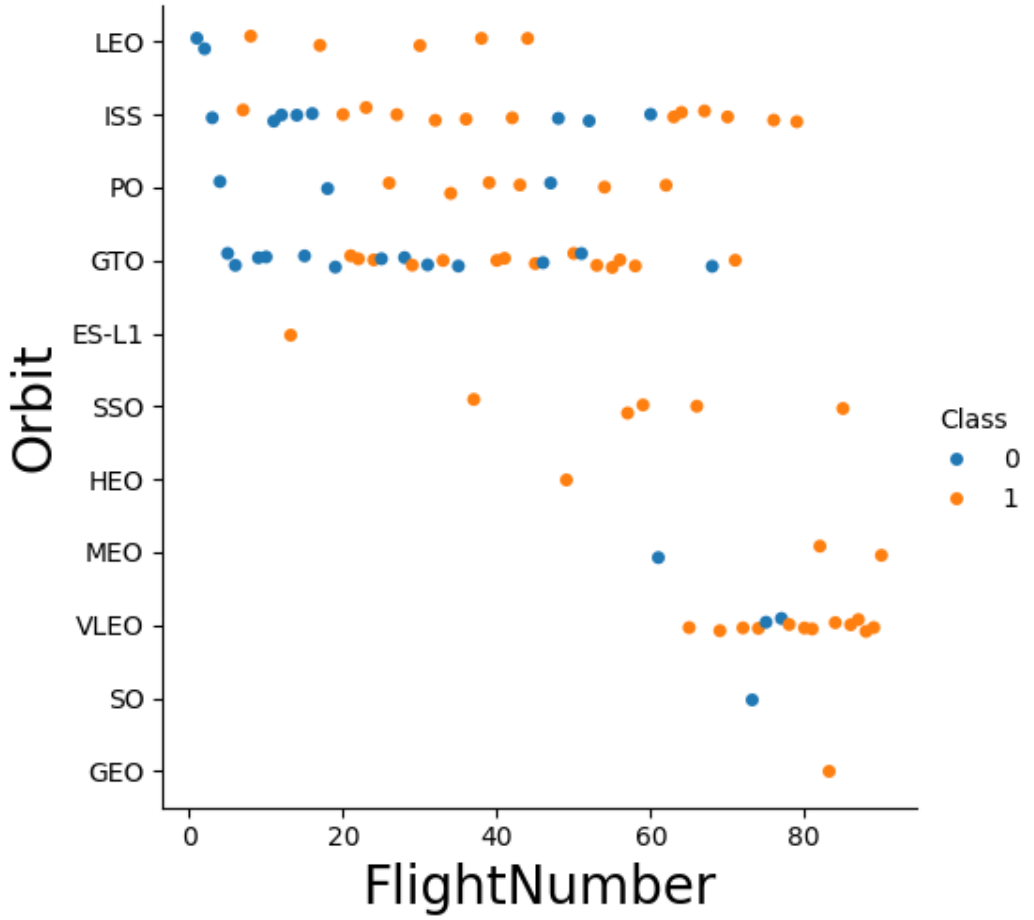
```
t = df.groupby(['Orbit', 'Class'])['Class'].agg(['mean']).reset_index()
sns.barplot(y="Class", x="Orbit", data=t)

plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Class",fontsize=20)
plt.show()
```



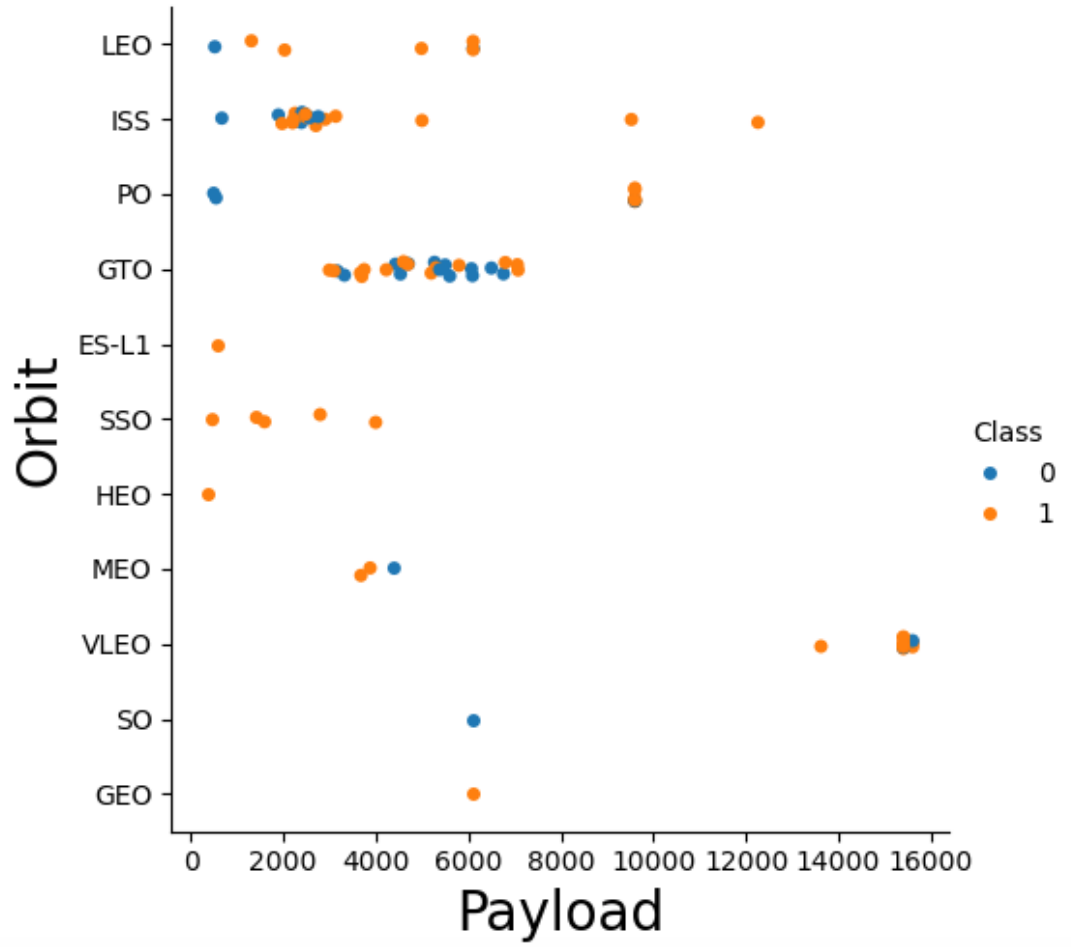
Flight Number vs. Orbit Type

```
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



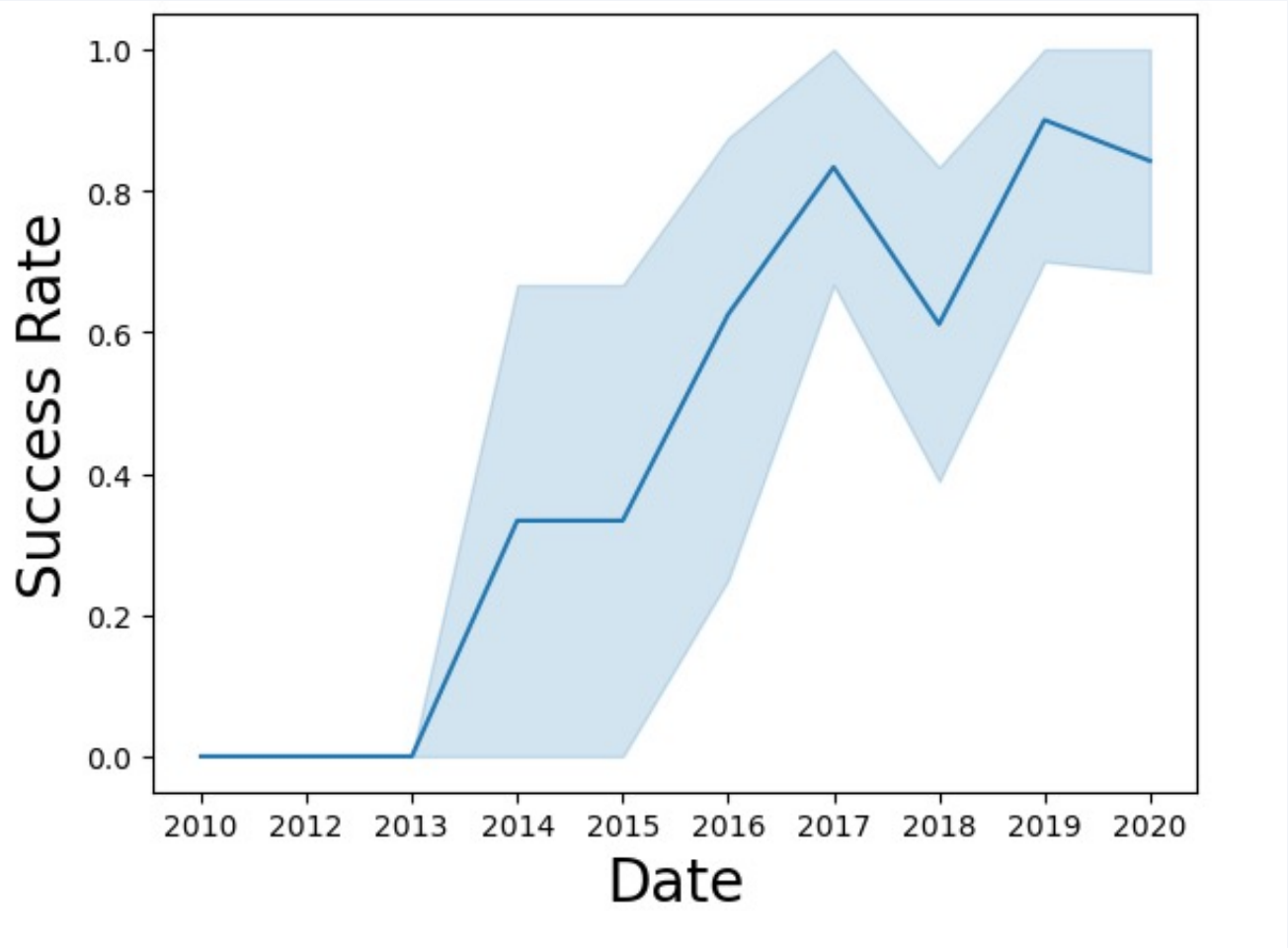
Payload vs. Orbit Type

```
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df)
plt.xlabel("Payload",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch Success Yearly Trend

```
sns.lineplot(data=df, x="Date", y="Class")
plt.xlabel("Date",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



All 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 Begin with 'CCA'

```
[10]: %sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
 * sqlite:///my_data1.db
Done.
```

Output View

```
 * sqlite:///my_data1.db
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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Total Payload Mass

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

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

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

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

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

```
[5]:
%sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'
```

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

[5]:

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

[16]:
```
%sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' \
    and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

[16]:

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

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)'
```

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

count(MISSION_OUTCOME)
99

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%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
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

```
[21]: %sql SELECT substr(min(DATE), 6, 2) AS month FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'

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

[21]:

month
12

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
```

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	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	4:45:00	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	5:21:00	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	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

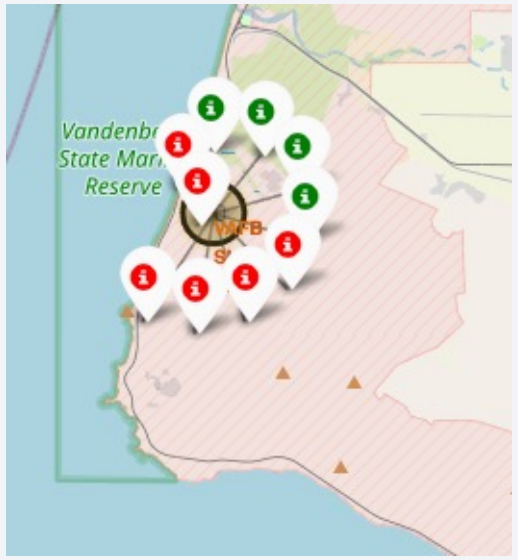

Section 3
Launch Sites
Proximities Analysis

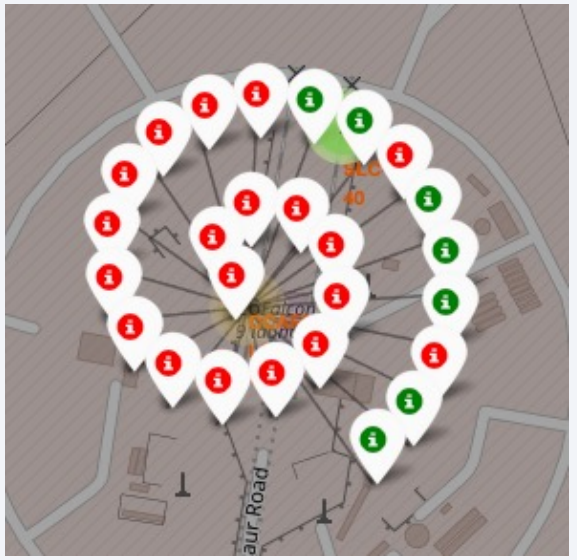
SpaceX US Launch Site Locations

Clustered in southern CA and FL, both clusters coastal

CA launch sites (top)
FL launch sites (bottom)

Successful/Failed Launch Markers





Left: Clusters in CA and FL indicating total launch attempt counts at each location
Middle: zoomed-in individual launch sites in CA, red icon indicating failure and green icon indicating success
Right: zoomed-in individual launch sites within one of several launch clusters within FL, red icon indicating failure and green icon indicating success

Mouse and Distance Markers

Section 4

Predictive Analysis (Classification)

Classification Accuracy

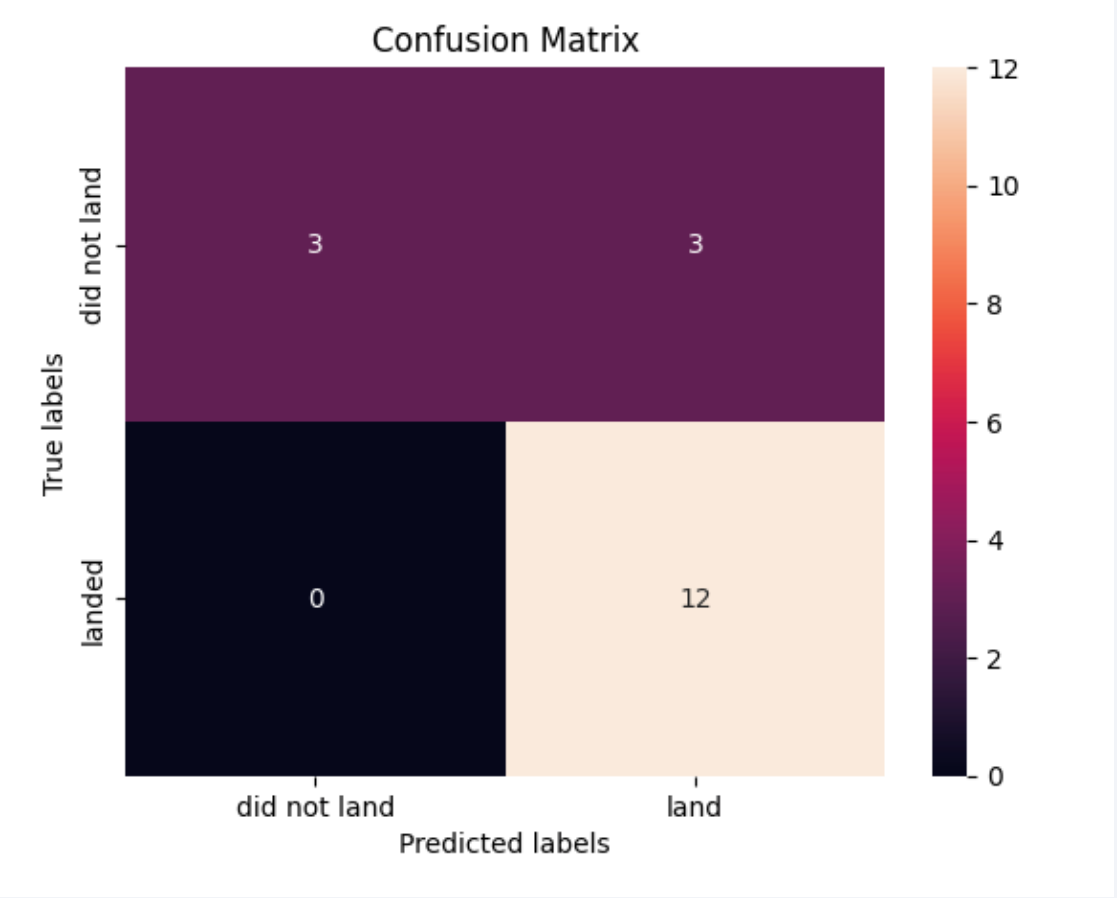
- 4 classification methods: logistic regression, support vector machine, decision tree, K-nearest neighbors
- Hyperparameter selection via grid search cross-validation

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm:',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Parameters:',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Parameters:',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params are:',logreg_cv.best_params_)
```

```
Best Algorithm: Tree with a score of 0.875
Best Parameters: {'criterion': 'entropy', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_sample
s_split': 10, 'splitter': 'random'}
```

Confusion Matrix

- Visualization of an accuracy metric (Type I (false positive), Type II (false negative) errors)
- Minimal Type II error



Conclusions

- Low-weighted payloads performed higher than heavier ones
- SpaceX launch success rates tend to improve over time, with the occasional phase of failure but a general upward trend
- The ES-L1, GEO, HEO, and SSO orbit types are generally the most successful
- The decision tree classification method served as the best predictor for launch success

Thank you!