IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yingling Yang
Sep 24 2021

Outline

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

Executive Summary

- Build, tune and evaluate different classification models to predict outcomes of Falcon 9 first stage landing based on data collection, data wrangling, exploratory data analysis and interactive visual analytics.

- Relationships between landing outcomes and different independent variables, such as launch sites, pay load, orbits and launch years, are investgated. Important factors affecting landing outcomes are determined by performing basic statistics and data visualization. All built models can successfully distinguish test samples between the different classes, and all the accuracies are above 0.8.

Introduction

By drastically reducing launch costs, SpaceX has revolutionized rocket technology in the 21st century. Falcon 9 from SpaceX only needs 6,2 million US dollars to complete the mission that other companies usually need 1.65 million dollars. The key to SpaceX ability to reduce launch costs is that the Falcon 9 rocket can be used repeatedly during the first launch stage. But the Falcon 9 rocket will not be successful every time, and this directly determines the final cost of the mission. The purpose of this project is to analyze the historical data of Falcon 9 launching to discover various factors that affect the landing success rate, and to build, train and evaluate different maching learning models to predict the probability of successful landing of Falcon 9 under various conditions. Such information is vital to other space companies competing with SpaceX.

Section 1

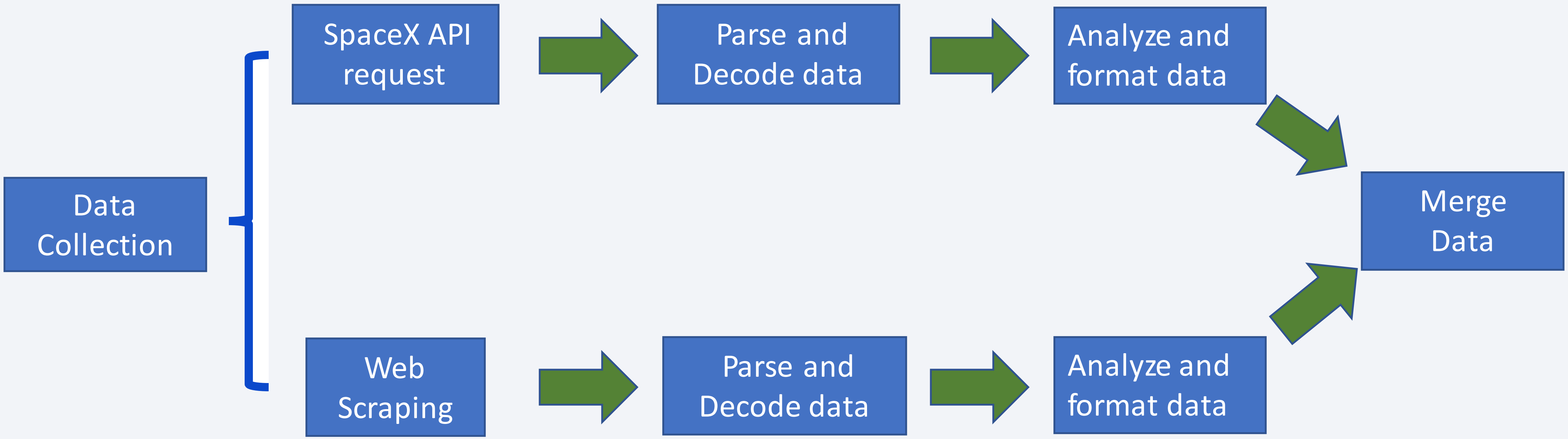
Methodology

Methodology

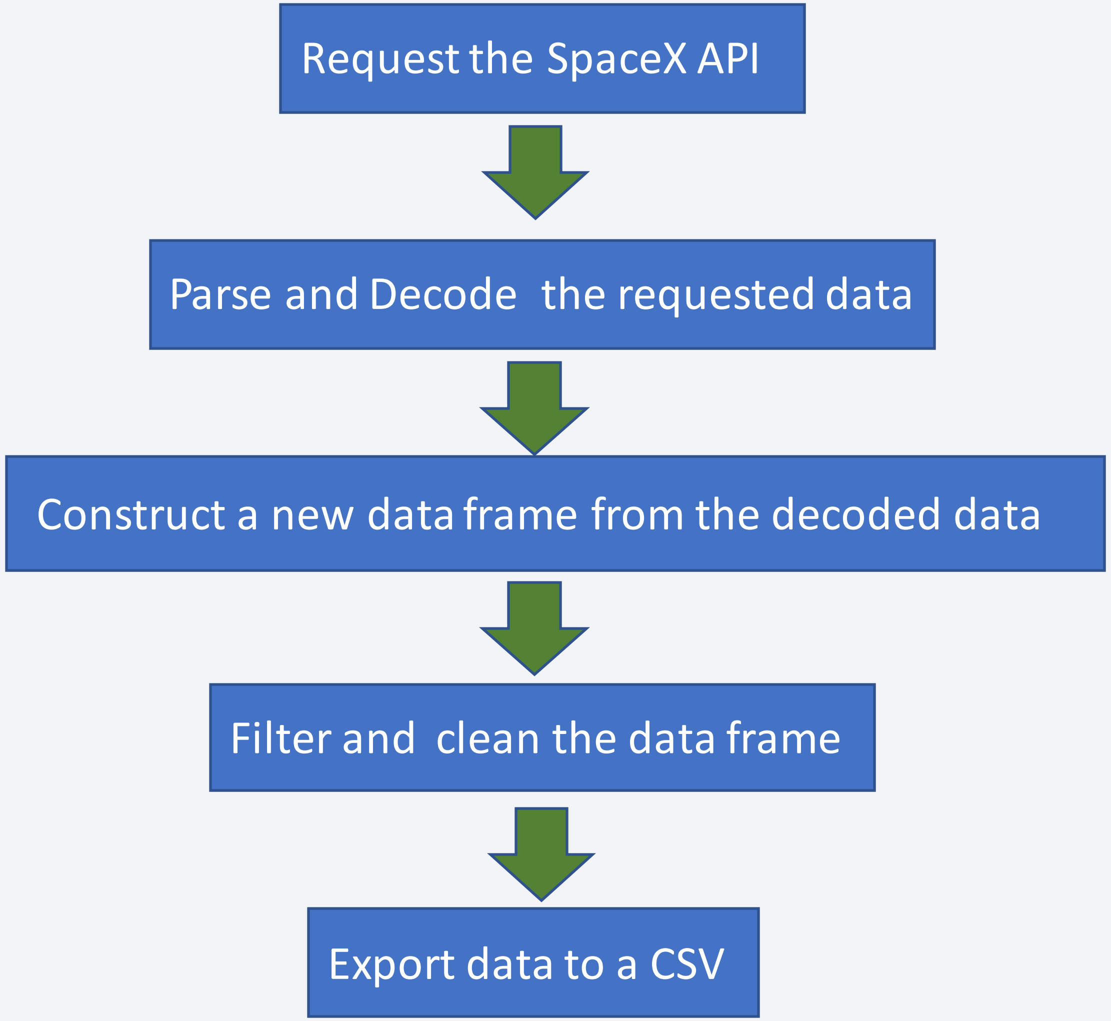
Executive Summary

- Data collection methodology:
  - SpaceX API request and web scrapling
- Perform data wrangling
  - Deal with smissing data and transform categorical data to numerical data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build logistic regression, SVM, decision tree and KNN models, use GridSearch to find best prameters for each models, use score method and coffusion matrix to evaluate models

Data Collection



Data Collection – SpaceX API



GitHub: https://github.com/yinglingyang/Data-Science-with-Python/blob/main/1-jupyter-labs-spacex-data-collection.ipynb

Data Collection - Scraping

Request the Wikipedia page

↓

Create a BeautifulSoup object

↓

Extract records from Wikipedia tables

↓

Parse the table and convert it into a Pandas data frame

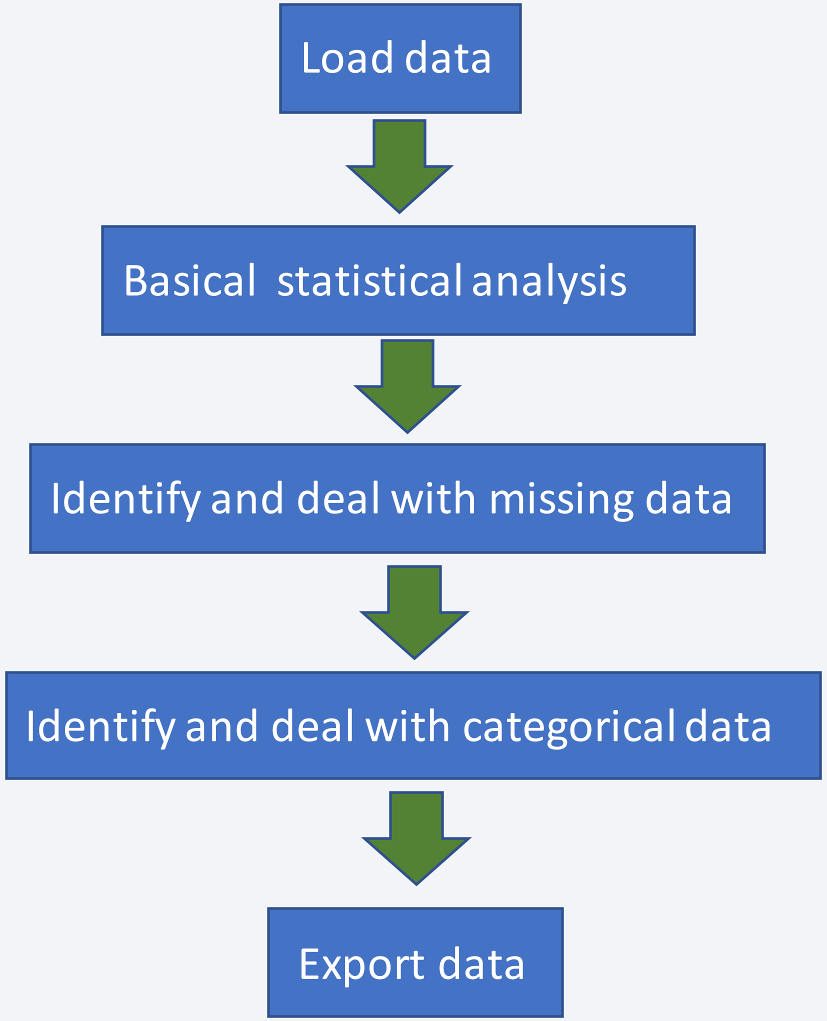
↓

Export data to a CSV

GitHub: https://github.com/yinglingyang/Data-Science-with-Python/blob/main/2-jupyter-labs-webscraping.ipynb

Data Wrangling

- Using df.isnull() to identify missing data
- Using df.dtype to identify categorical data
- Drop or replace missing data
- Transform categorical data to numerical data
- Using df.value_counts(), df.count(), df.sum and df.describle() to do basical statistical analysis



GitHub: https://github.com/yinglingyang/Data-Science-with-Python/blob/main/3-jupyter-spacex-Data-wrangling.ipynb

EDA with Data Visualization

To understand relationships between variables and determine which variables for machine training, those charts are plotted:

- Scatter plot of FlightNumber vs. LaunchSite
- Scatter plot of Payload vs. Launch Site
- Bar plot of Orbit type vs. Success rate
- Scatter plot of FlightNumber and Orbit type
- Scatter plot of Payload and Orbit type
- Line plot of Year and Success rate

GitHub: https://github.com/yinglingyang/Data-Science-with-Python/blob/main/5-jupyter-labs-eda-data-Visualization.ipynb

EDA with SQL

Performed SQL queries:

- *Display the names of the unique launch sites in the space mission*
- *Display 5 records where launch sites 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 achieved*
- *List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000*
- *List the total number of successful and failure mission outcomes*
- *List the names of the booster_versions which have carried the maximum payload mass. Use a subquery*
- *List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015*
- *Rank 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¶*

GitHub: https://github.com/yinglingyang/Data-Science-with-Python/blob/main/4-jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- Add folium.Circle and folium.map.Marker for each launch site to see where these launch sites are located
- Add folium.plugins.MarkerCluster() for each launch site to mark the success/failed launches and see which sites have high success rates
- Add MousePosition to get the coordinates and calculate the distances between a launch site to its proximities
- Add folium.Marker and folium.PolyLine to show the distances between a launch site to its proximities

GitHub: https://github.com/yinglingyang/Data-Science-with-Python/blob/main/6-jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Add a dropdown list to enable launch site selection
- Add the piechart of success count for all sites to show which site has the largest successful launches and the piechart for each site to show which one has the highest launch success ratio
- Add a slider to enable payload range selection
- Add a scatter chart of Payload Mass (kg) vs. Class to show the correlation between payload and launch success

GitHub: https://github.com/yinglingyang/Data-Science-with-Python/blob/main/7-spacex_dash_app.py

Predictive Analysis (Classification)

Load data and use preprocessing.StandardScaler() to standardize the data

↓

Use train_test_split to split the data into training and test data sets

↓

Bulid Logistic Regression, SVM, Decision Tree and KNN models, then use GridSearchCV to fit train data set and find Best parameters

↓

Using score method and confusion matrix to evaluate the accuracies of models on the test data set

↓

Compare results and find the best model

GitHub: https://github.com/yinglingyang/Data-Science-with-Python/blob/main/8-Prediction.ipynb

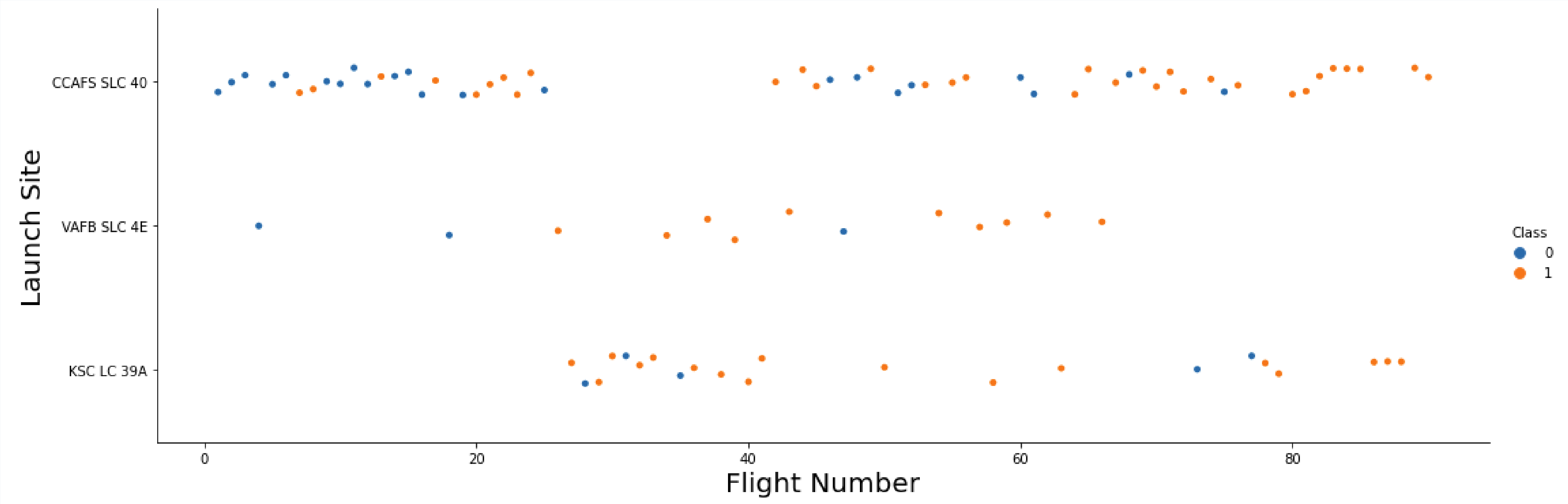
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

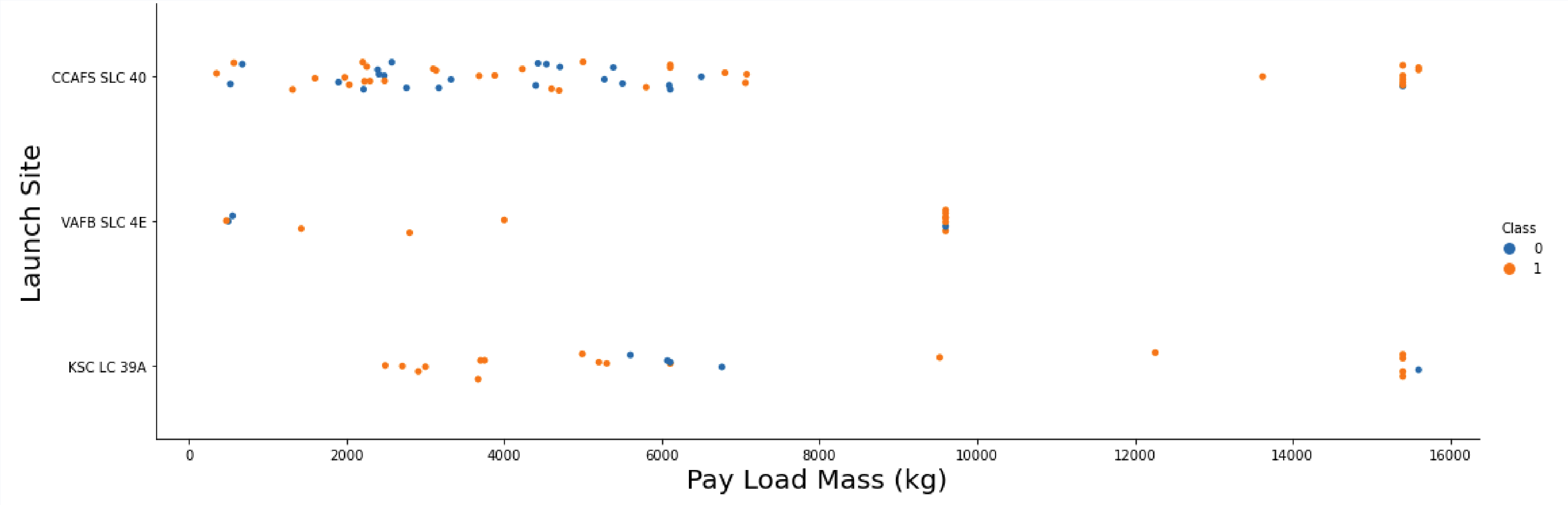
Insights drawn from EDA

Flight Number vs. Launch Site



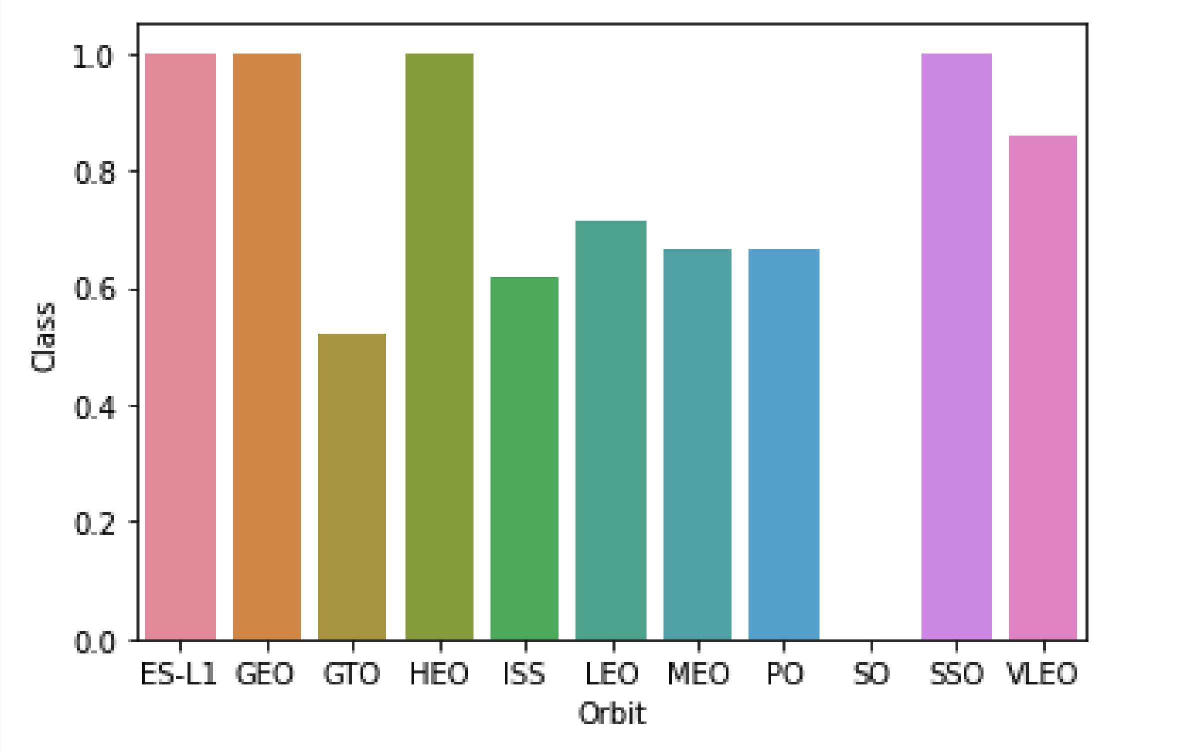
- As the flight number increases, the success rate of landing increases
- Different launch sites have different success rates
- CCAFS SLC 40 has the lowest success rate

Payload vs. Launch Site



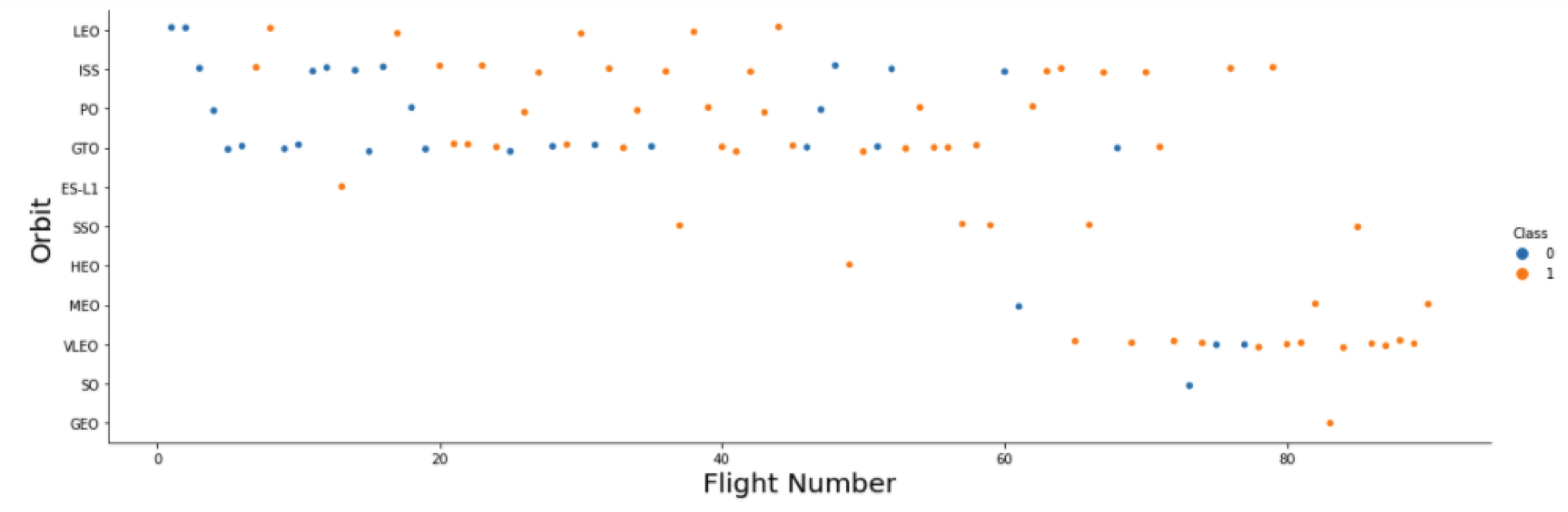
- The success rate is lowest when pay load mass is around 6000 kg
- All the pay load mass are less than 10000 kg on VAFB SLC 4E launch site
- Pay load mass can be greater 10000 kg on both CCAFS SLC 40 and KSC LC 39A sites.

Success Rate vs. Orbit Type



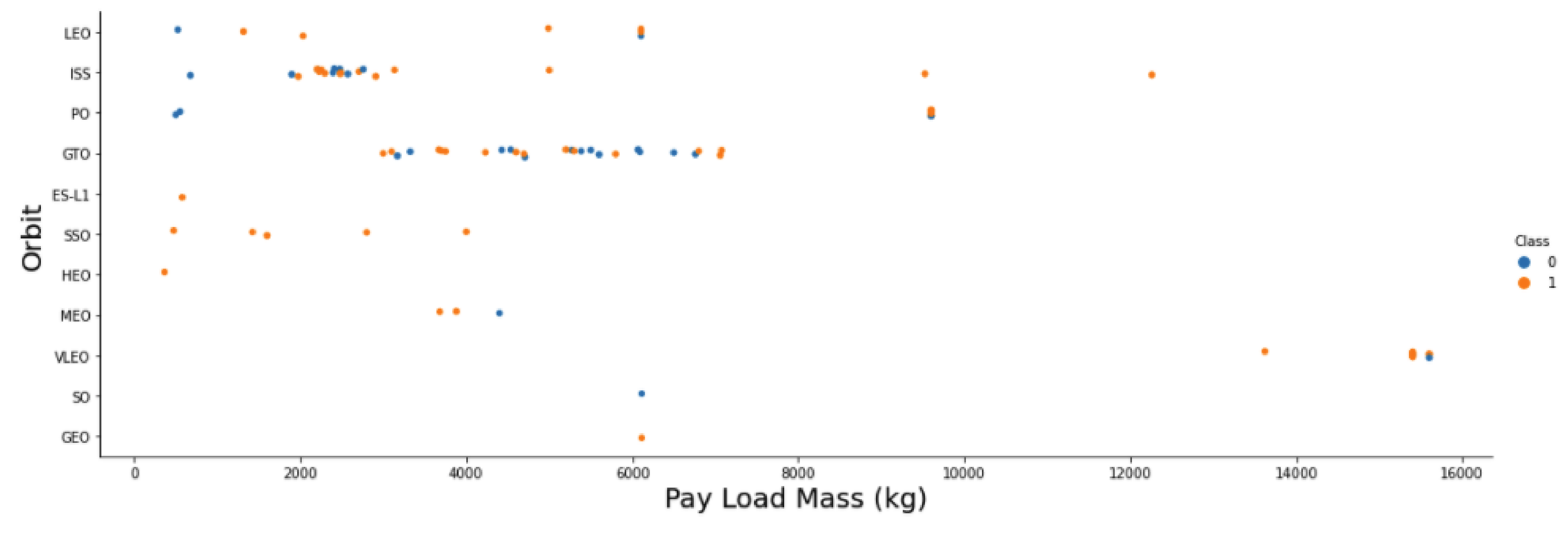
- ES-L1, GEO, HEO and SSO have the highest success rates (about 1)
- GTO has the lowest success rates (about 0.5)

Flight Number vs. Orbit Type



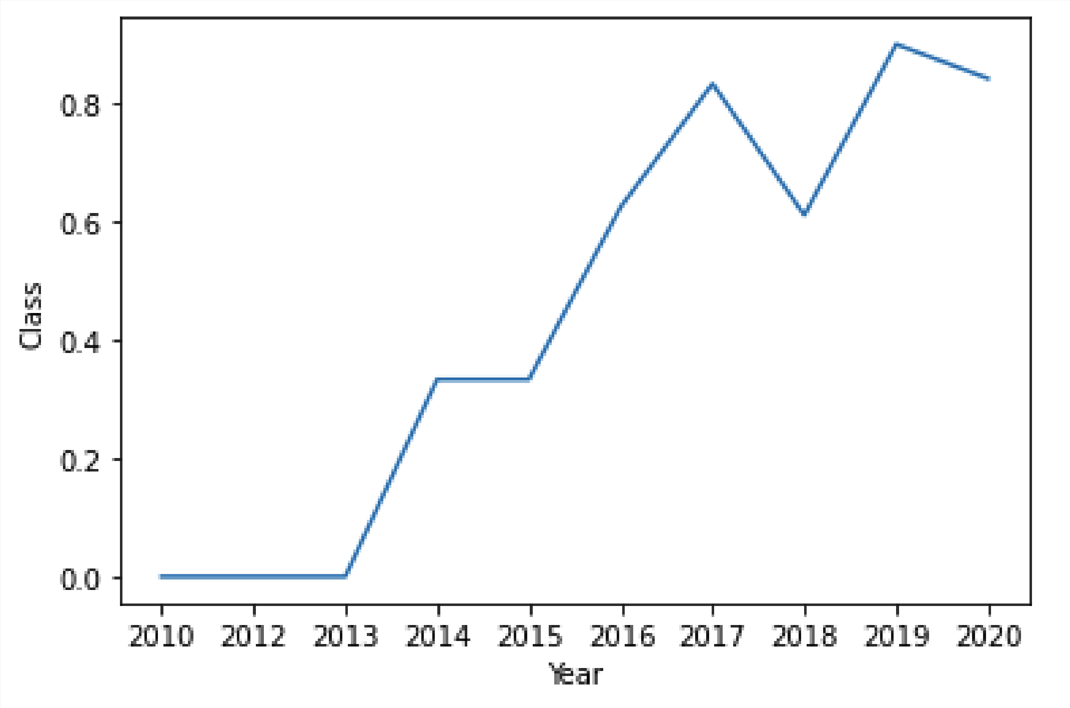
- For LEO, PO and MEO orbits, the success rate seems increases with flight number

Payload vs. Orbit Type



- Heavy payloads have a negative influence on GTO, MEO and VLEO orbits
- Heavy payloads have a positive influence on ISS orbits

Launch Success Yearly Trend



The success rate of landing increases with year since 2013

All Launch Site Names

```
%%sql

select distinct LAUNCH_SITE
from SPACEXDATA
```

```
 * ibm_db_sa://wtp89840:***@fbd88901
Done.
```

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

Query result: CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E

Launch Site Names Begin with 'CCA'

Query result:

```
%%sql

select *
from SPACEXDATA
where LAUNCH_SITE like 'CCA%'
limit 5
```

```
 * ibm_db_sa://wtp89840:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/BLUDB
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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

```
%%sql

select sum(payload_mass__kg_)
from SPACEXDATA
where CUSTOMER like '%NASA%CRS%'
```

```
 * ibm_db_sa://wtp89840:***@fbd88901-
32731/BLUDB
Done.
```

1
48213

The total payload carried by boosters from NASA (CRS) is 48213

Average Payload Mass by F9 v1.1

```
%%sql

select avg(payload_mass__kg_)
from SPACEXDATA
where booster_version like 'F9 v1.1%'
```

```
 * ibm_db_sa://wtp89840:***@fbd88901-ebdb-
Done.
 1
2534
```

Query result: 2534

First Successful Ground Landing Date

```
%%sql

select min(DATE)
from SPACEXDATA
where landing__outcome like '%Success%ground pad%'
```

```
 * ibm_db_sa://wtp89840:***@fbd88901-ebdb-4a4f-a32e-982
Done.
```

1
2015-12-22

Query result: 2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql

select distinct booster_version
from SPACEXDATA
where (landing__outcome like '%Success%drone ship%')
      and (payload_mass__kg_ between 4000 and 6000)
```

```
 * ibm_db_sa://wtp89840:***@fbd88901-ebdb-4a4f-a32e-9822b
Done.
```

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

Query result: F9 FT B1021.2, F9 FT B1031.2, F9 FT B1022, F9 FT B1026

Total Number of Successful and Failure Mission Outcomes

```
%%sql

select count(*) as number_success
from SPACEXDATA
where mission_outcome like '%Success%'
```

```
* ibm_db_sa://wtp89840:***@fbd88901-ebdb-4
Done.
```

number_success
100

```
%%sql

select count(*) as number_failure
from SPACEXDATA
where mission_outcome like '%Failure%'
```

```
* ibm_db_sa://wtp89840:***@fbd88901-ebdb-4
Done.
```

number_failure
1

Query result: number of success is 100 and number of failure is 1

Boosters Carried Maximum Payload

```
%%sql

select distinct booster_version
from SPACEXDATA
where payload_mass__kg_ in (select max(payload_mass__kg_)
                            from SPACEXDATA)
```

```
 * ibm_db_sa://wtp89840:***@fbd88901-ebdb-4a4f-a32e-9822b9fb23
Done.
```

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

2015 Launch Records

The failed landing in drone ship, their booster versions, and launch site names in year 2015:

```
%%sql

select booster_version, launch_site
from SPACEXDATA
where DATE like '%2015%'
      and landing__outcome like '%Failure%drone ship%';
```

```
 * ibm_db_sa://wtp89840:***@fbd88901-ebdb-4a4f-a32e-9822b9fb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

```
%%sql

SELECT landing__outcome, count(*) as landing
FROM SPACEXDATA
WHERE DATE between '2010-06-04' and '2017-03-20'
GROUP BY landing__outcome
ORDER BY landing desc;
```

```
 * ibm_db_sa://wtp89840:***@fbd88901-ebdb-4a4f-a32e-
Done.
```

landing__outcome	landing
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

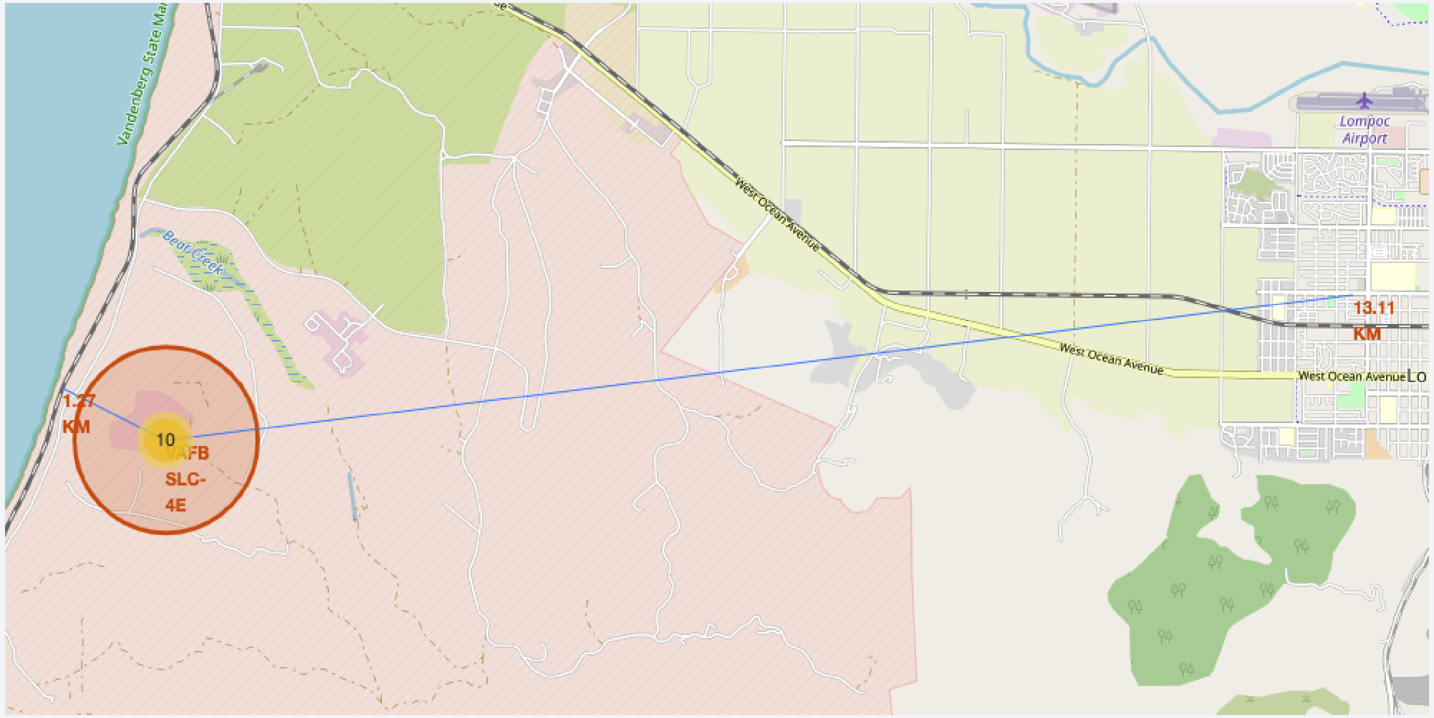
Locations of All Launch Sites

- All launch sites are close to the Equator line
- All launch sites are very close to the coast line

Success/Failed Launches for Each Site

- Green marker for successful landing and red marker for failed landing
- From the color-labeled markers in marker clusters, we can identify CCAFS SLC-40 launch site has relatively high success rates.

Distances between Launch Sites to Their Proximities

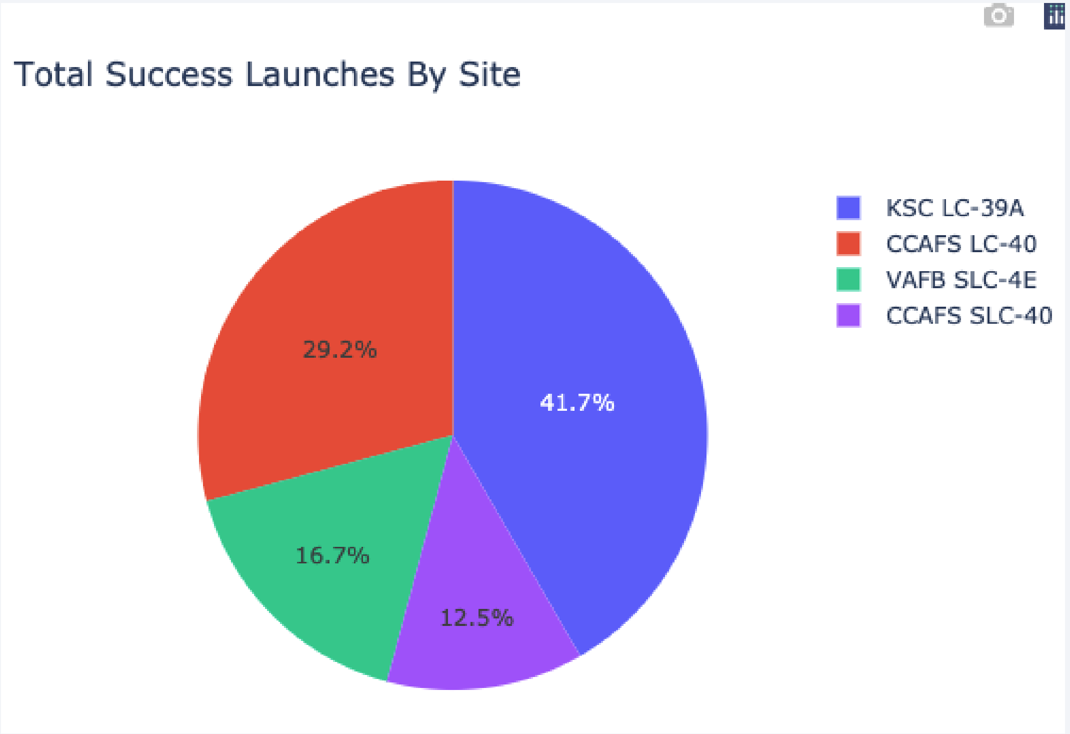


- All launch sites are in close proximity to railways, highways and coastline
- All launch sites keep certain distance away from cities

Section 5

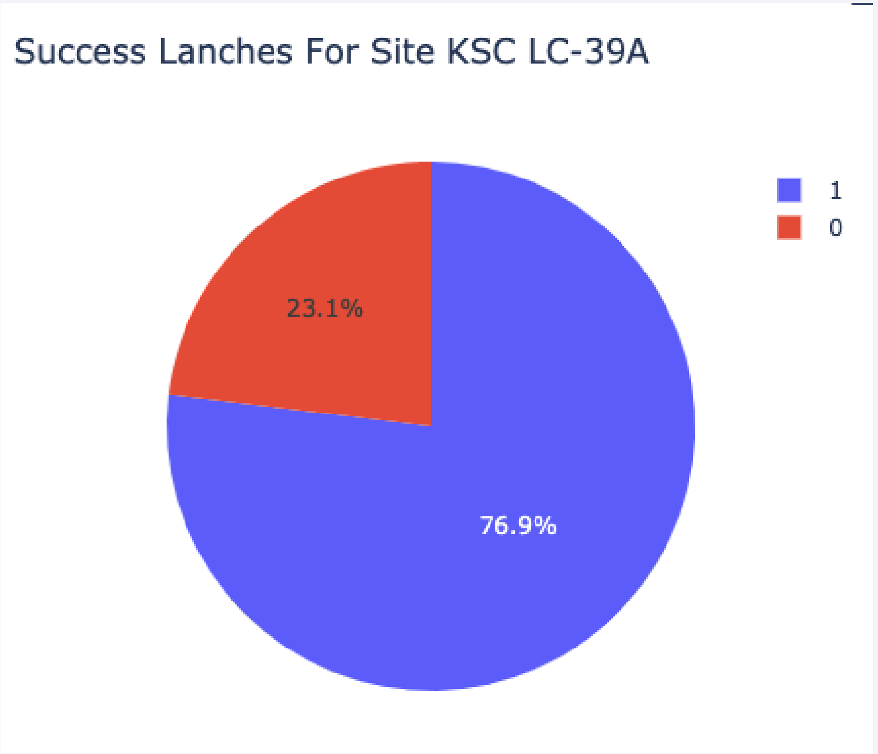
Build a Dashboard with Plotly Dash

Total Success Launches by Site



- KSC LC-39A site has the largest successful launches
- VAFB SLC-4E site has the smallest successful launches

Success Launch Rate for Each Site



Launch Success Rate:

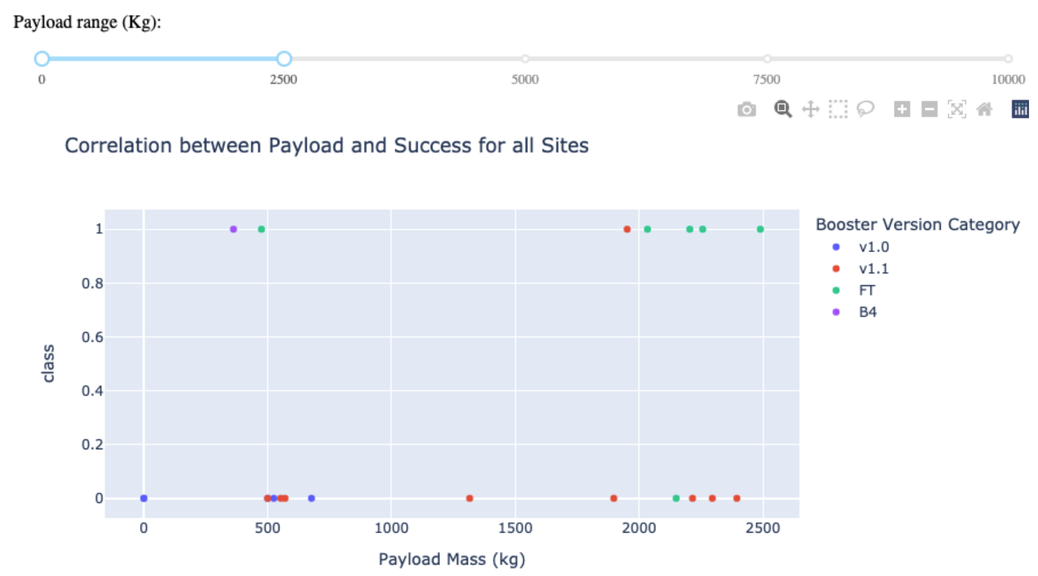
- CCAFS LC-40 (26.9%)
- VAFB SLC-4E (40%)
- KSC LC-39 A (76.9%)
- CCAFS SLC-40 (42.9%)

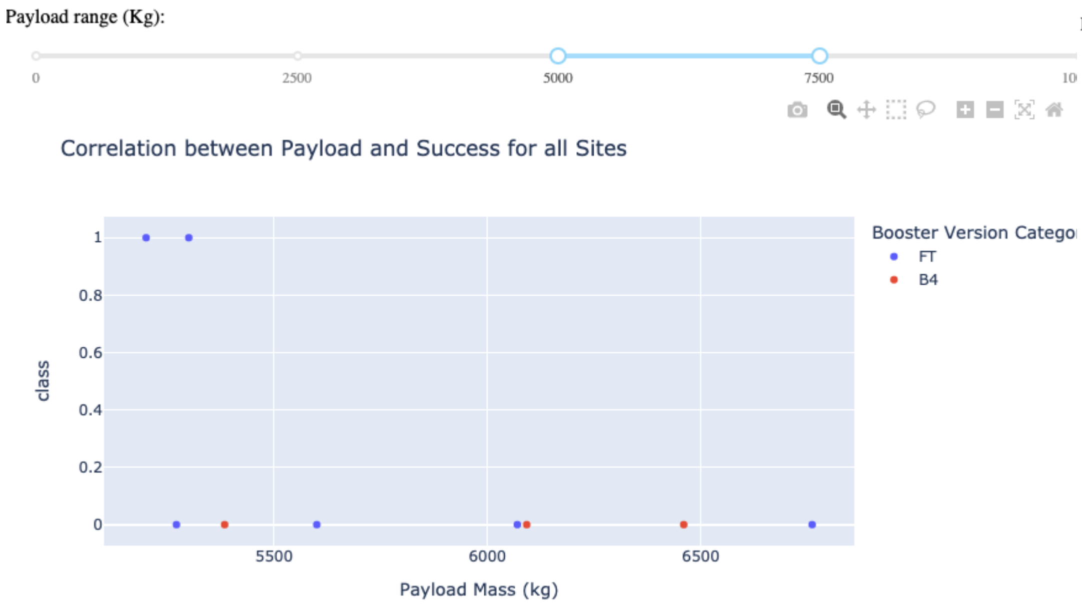
KSC LC-39 A (76.9%) has the highest launch success rate.

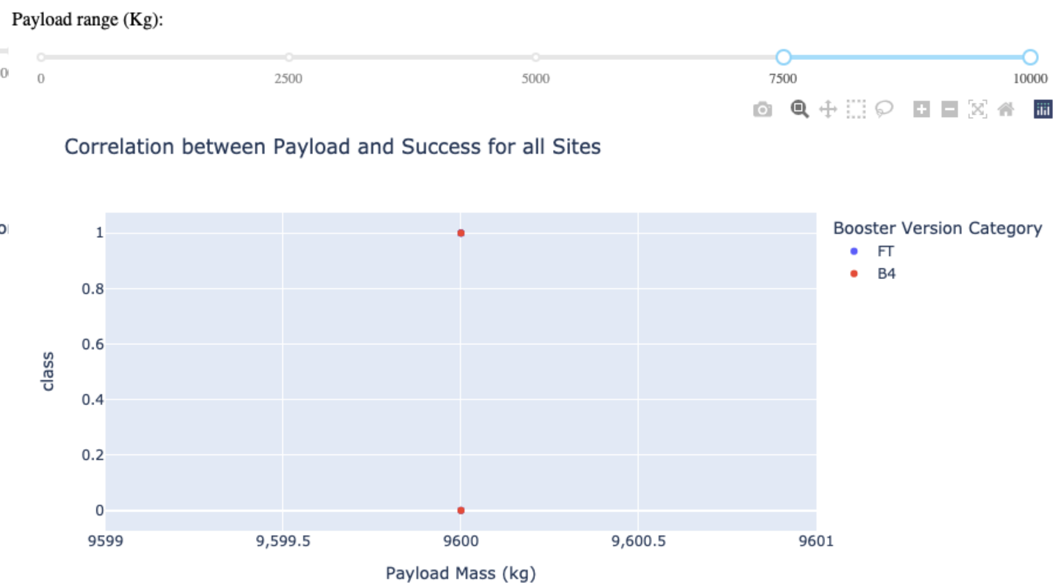
Correlation Between Payload and Success

- Payload range(2000-4000) has the highest launch success rate
- Payload range(6000-8000) has the lowest launch success rate
- Booster version B5 has the highest launch success rate

Correlation Between Payload and Success



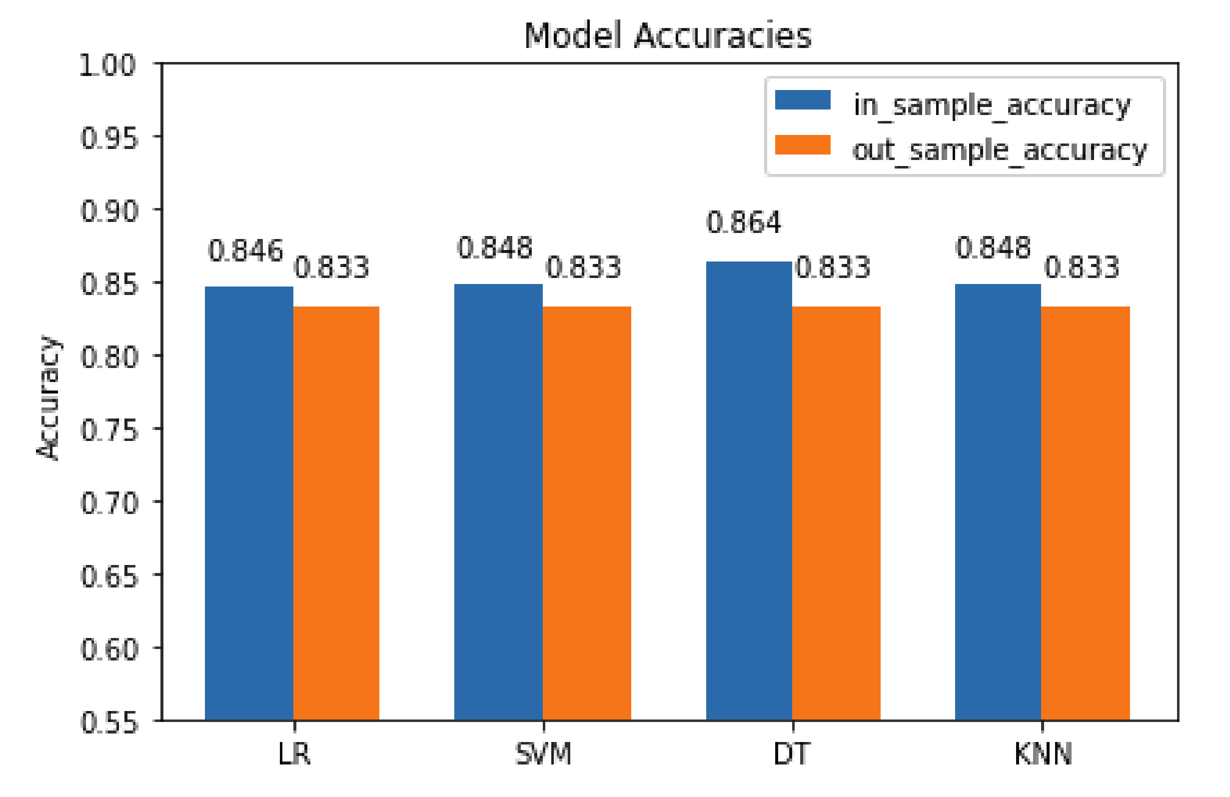




Section 6

Predictive Analysis (Classification)

Classification Accuracy



- All models have good in sample accuracies and out sample accuracies (>0.8)
- Dicision Tree Model has the best in sample accuracy
- All models have simlar out sample accuracies

Confusion Matrix

- All models can distinguish test samples between the different classes
- The major problem of these models is false positives

Conclusions

- The data analysis and visualization show that the success rate of Falcon 9 first stage landing is affected by many factors, including launching date, launching site, payload and orbit.
- Four classification models were built, tuned and evaluated after data cleaning, data analysis and visualization. These models are successful predict the outcome for test data with accuracies above 0.8.

Appendix

GitHub LinK:

https://github.com/yinglingyang/Data-Science-with-Python

Thank you!