IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Bartłomiej Janas
02.03.2024

Outline

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

Executive Summary

- Methodologies for this project will tackle most of problems that emerge from this data and visualize the most important aspects.
- Achieved result is that mission outcome can be predicted while having appropriate attributes and suffice size of data.

Introduction

- Main goal of this project is to help company named SpaceY - lead by Allon Mask - to decide if it is worth investing into reusable first stages for rockets.
- Analysis will be made on past missions of similarly named company: SpaceX, which completed numerous launches with reusable first stages (with some of them coming back in one piece, some of them not). Focus will be on Falcon 9 booster.
- If we compare prices from different companies for transporting around 21 tonnes to LEO orbit, we can see that Atlas V costs $158 million, Ariane 5: $137 million, where Falcon 9: $57 million.
- Reusability of first stage (up to 7 times) greatly contributes to lowering the cost of the mission and gaining places in World Space Race. Thus gaining insights for predicting which Falcon 9 booster will safely return home and which factors determine the mission status could yield valuable business information for SpaceY.

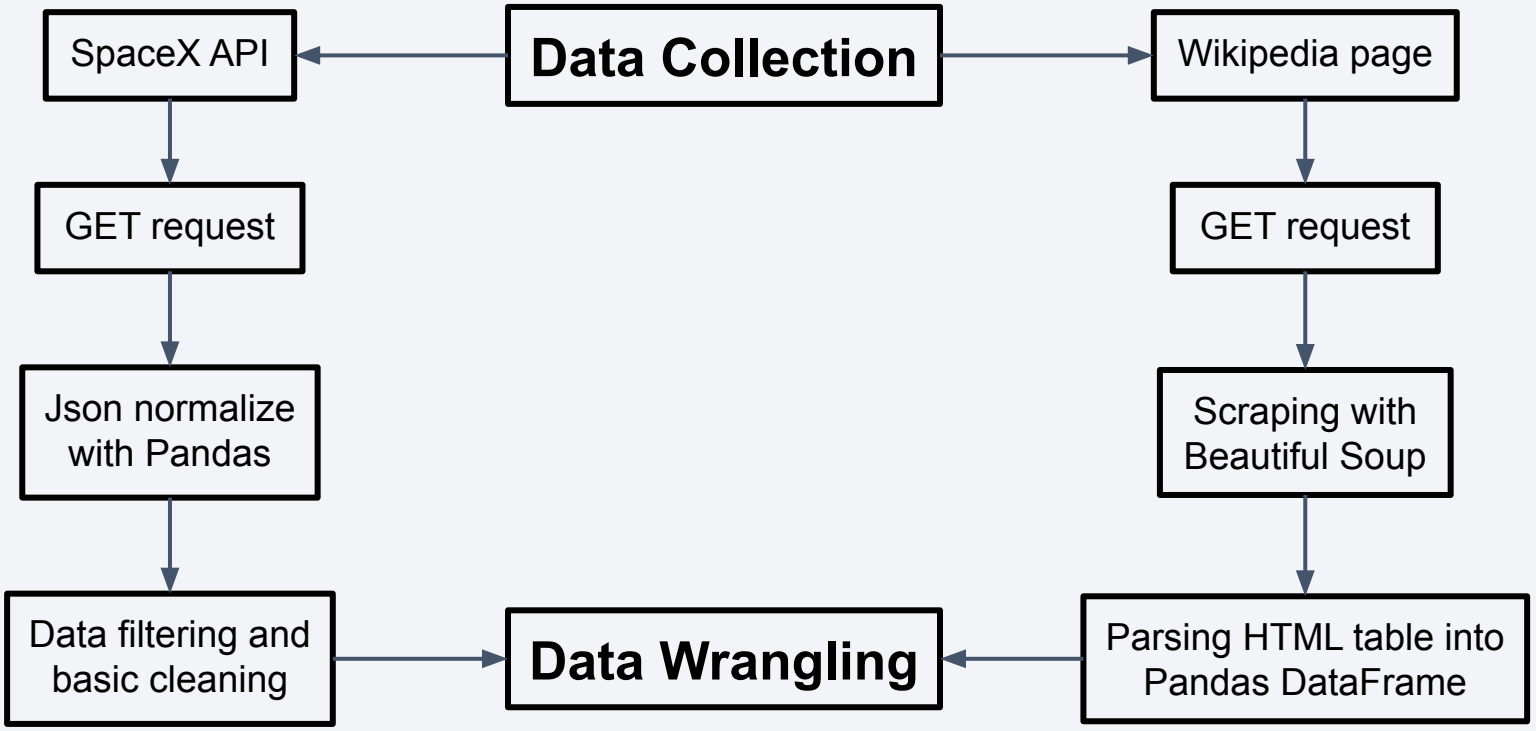
Section
1

Methodology

Methodology

- Data collection methodology:
  - Collection using SpaceX Web API and scraping historical launch records
- Perform data wrangling
  - Transforming and cleaning data to separate successful and unsuccessful launches with clear numerical-encoded features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning and evaluating models as: Logistic Regression, Support Vector Machines, Decision Tree Classifier and K-Nearest Neighbors

Data Collection



Data Collection – SpaceX API

1. Acquiring response as GET request from url: “https://api.spacexdata.com/v4/launches/past”
2. Normalizing json response for Pandas DataFrame
3. Filtering for the most useful features
4. Creating empty dictionary and filling it up by a for loops on previous data
5. Dictionary to DataFrame and filtering to only have Falcon 9 booster
6. Filling missing Payload values
7. Exporting data to CSV

```
requests.get(spacex_API_url)
```

```
data = pd.json_normalize(response.json())
```

```
data[data['BoosterVersion'] == 'Falcon 9']
```

```
data['PayloadMass'].fillna(data['PayloadMass'].mean(), inplace=True)
```

```
data.to_csv('dataset_part_1.csv', index=False)
```

[Data Collection Jupyter notebook]

Data Collection - Scraping

1. Acquiring response as GET request from wikipedia url
2. Creating a BeautifulSoup object form the HTML response
3. Collecting column names from HTML table header
4. Parsing launch HTML tables to a custom dictionary
5. Converting dictionary to Pandas DataFrame
6. Exporting DataFrame as CSV file

```
requests.get(static_wikipedia_url)
```

```
soup = BeautifulSoup(response.text)
```

```
launch_table = soup.find_all('table')[2]
```

```
for loop on launch_table to fill in launch_dict
and convert it to DataFrame df
```

Web-scraping Jupyter notebook

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

1. Loading in previous data
2. Basic exploring of missing values, data types
3. Comparing different launch sites and orbits
4. Filtering the outcomes for clear successes and failures
5. Creating binary classification target label where 1 indicates successful outcome of mission and 0 - unsuccessful outcome
6. Saving data to a CSV file for EDA

```
df = pd.read_csv('data.csv')
```

```
bad_outcomes=set(landing_outcomes.
keys()[[1,3,5,6,7]])
```

```
df['Outcome'].apply(lambda x:
0 if x in bad_outcomes else 1)
```

```
df.to_csv("dataset_part_2.csv",
index=False)
```

Data Wrangling Jupyter notebook

EDA with Data Visualization

- Categorical plot with: x=’FlightNumber’, y=’PayloadMass’, hue=’Class’. Insight into how payload affect mission outcome across continuous launch attempts
- Categorical plot with: x='FlightNumber', y='LaunchSite', hue='Class'. Which launch sites have the highest success rates and how it varied across continuous launch attempts
- Scatter plot with: x='PayloadMass', y='LaunchSite', hue='Class'. Analysing payload range between launch sites and its effect on mission outcome
- Bar plot with: x='Orbit' and mean of y='Class'. Comparing success rates between missions lead to different orbits
- Categorical plot with: x='FlightNumber', y='Orbit', hue='Class'. Finding relationship between orbit and mission outcome across continuous launch attempts
- Scatter plot with: x='PayloadMass', y='Orbit', hue='Class'. Looking into patterns into mission outcome with different types of orbit and payload masses
- Line plot with: x='Date' and mean of y='Class'. Visualizing timeline of success rate

EDA Data Visualization Jupyter notebook

EDA with SQL

Summary of performed queries:

- Displaying names of unique launch sites
- Selecting 5 records where launch sites begin with ‘CCA’
- Showing the total payload mass carried by boosters by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in a ground pad was achieved
- Presenting the names of boosters which have successfully landed in a drone ship and have payload mass between 4000 and 6000 kgs
- Selecting the total number of successful and failure mission outcomes
- Listing the names of booster versions which have carried the maximum payload mass
- Displaying months, failure landing outcomes in drone ship, booster versions and launch sites for the year 2015
- Ranking total number of different landing outcomes between 2010-06-04 and 2017-03-20

EDA SQL Jupyter notebook

Build an Interactive Map with Folium

Added objects to a map:

- Circle at NASA Johnson Space Center to visualize its position regarding launch sites
- Circle and marker at each launch site to show from which parts of USA are SpaceX rockets starting
- Marker cluster for each mission from launch sites indicating the outcome with green or red color
- Mouse Position for further getting coordinates of various places near launch sites
- Marker and Polyline to the closest coastline from SLC_40. Analyzing the proximity of different landmarks for safety reasons, the same goal for last 3 objects
- Marker and Polyline to the closest railway from SLC_40
- Marker and Polyline to the closest highway from SLC_40
- Marker and Polyline to the closest city from SLC_40

Launch Site Map Jupyter notebook
Viewable Online Maps

Build a Dashboard with Plotly Dash

Graphs and interactions added to a dashboard:

- Dropdown menu to select or search for certain launch site or choosing all of them
- Range slider to choose specified Kg range for payload mass, from 0 to 10000 with a step of 1000
- Pie chart indicating success rate for selected launch site, or comparing all of them. Interactive dropdown menu lets user choose what should be rendered on the pie chart
- Scatter plot showing outcomes of missions across payload mass. Once again user is able to specify which launch site should be displayed on the plot and what range of payload mass is he interested into. Both interactive menus change the layout of the plot

Plotly Dashboard Python file
CSV file for Dashboard

Predictive Analysis (Classification)

1. Standardizing independent variables with sk-learn's StandardScaler
2. Splitting data into 80% train and 20% test sets
3. Creating a dictionary of possible parameters for each used model: Logistic Regression, Support Vector Machines, Decision Tree Classifier and K-Nearest Neighbors
4. Grid Searching through parameters with cross-validation across 10 folds to find the best hyperparamaters for 4 models on training set
5. Fitting on training set and evaluating tuned models on test set to get accuracy score
6. Visualizing confusion matrix and comparing performance of every model

[ML Prediction Jupyter notebook]

Standardizing and train/test splitting the data

Choosing a model: LogReg, SVM, TreeCl, KNN

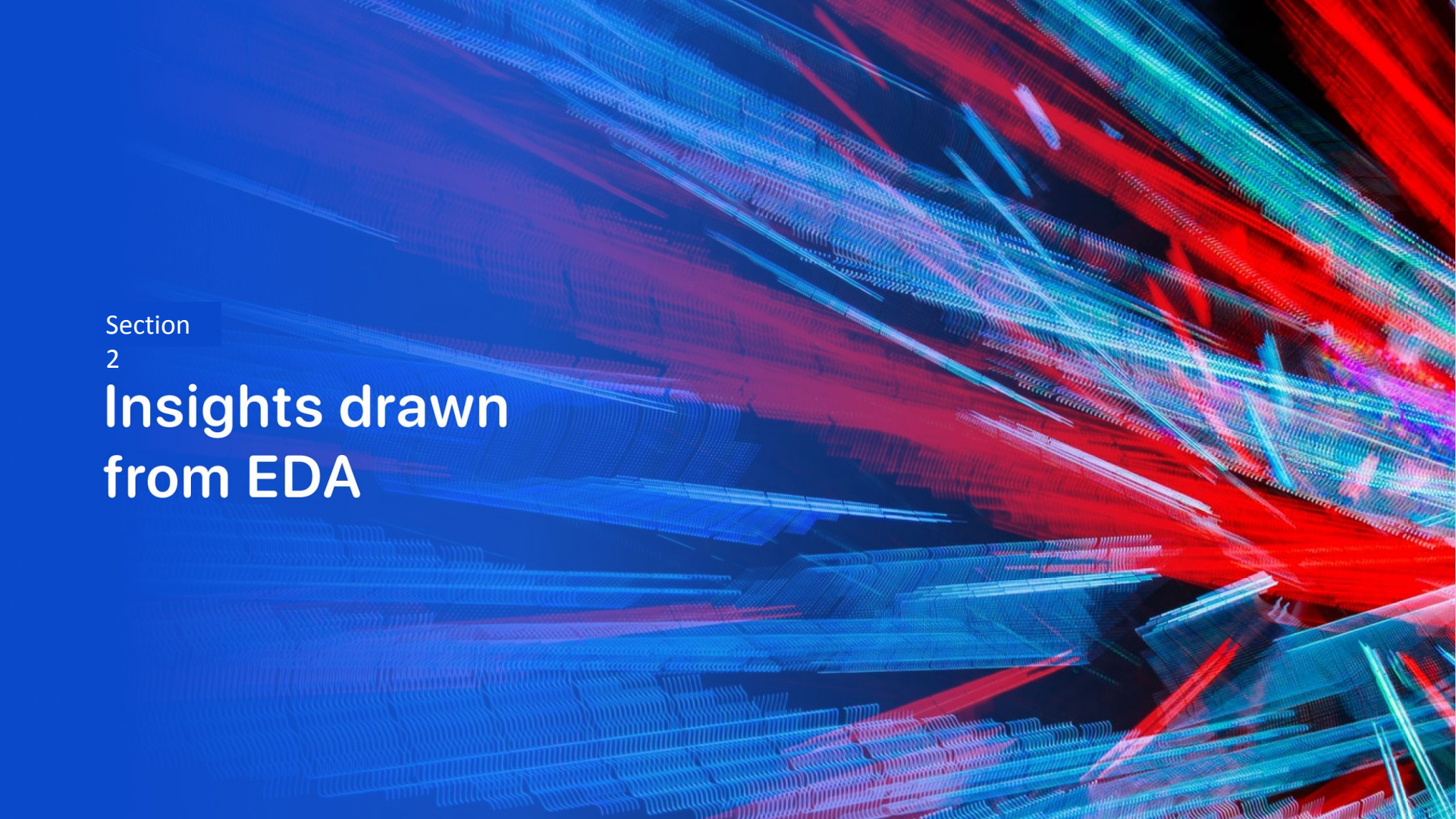
GridSearchCV with a range of parameters and 10 folds

Evaluating tuned model on a test data

Choosing the best model

Results

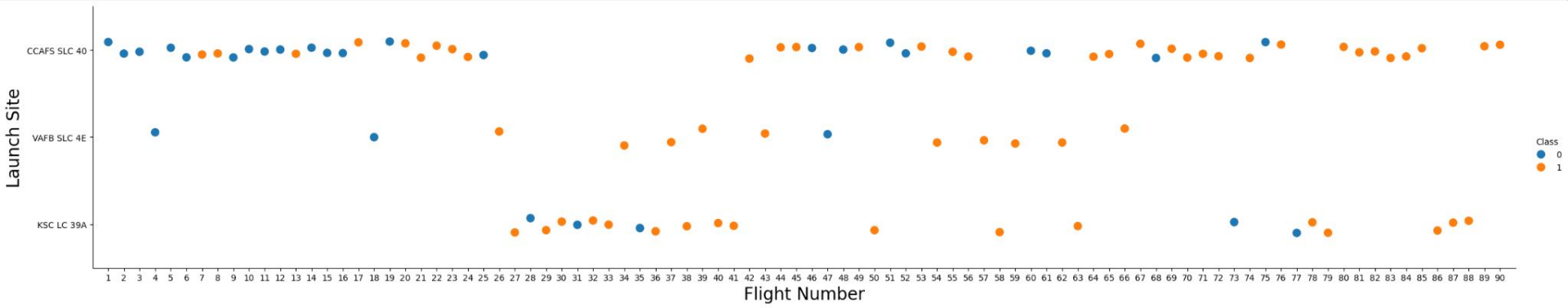
- EDA provided valuable information about the data. Analysing the differences between data across multiple features and time yielded insights in this even small (90 samples) data set
- Interactive analytics provided a dashboard and maps. Both of these features where crafted with the goal to show meaningful information to a researcher and other readers
- Predictive analysis proved that when having sufficient information about the upcoming launch, it is possible to predict its outcome, thus helping to prevent unsuccessful landings and saving funds on reusable boosters



Section 2

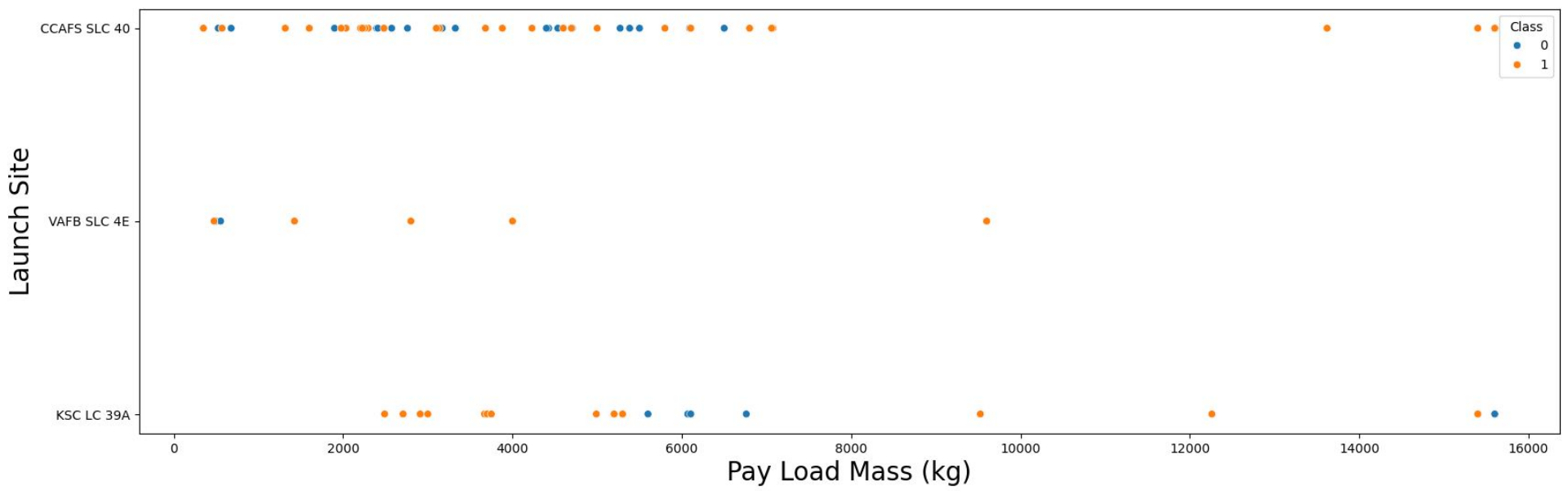
Insights drawn from EDA

Flight Number vs. Launch Site



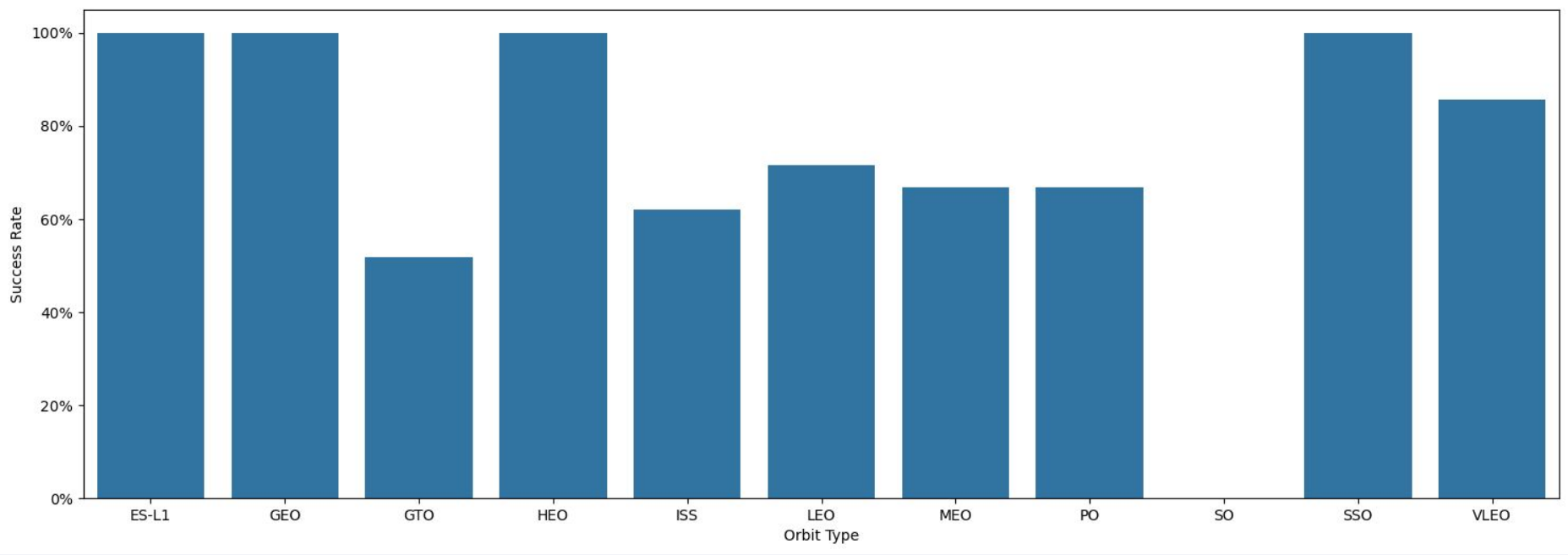
Although VAFB SLC 4E has really high success rate, not many rockets started there in comparison to other launch sites. It can be seen that the site of CCAFS SLC 40 not only has the most missions, but also it looks like it has achieved consistency for later missions. Many unsuccessful records transfer across launch sites, forming small vertical clusters. For example the failure of flight number 75 can be partially explained by flights 73 and 77. Pointing into an issue not really related to the launch site, so the last blue dot for SLC 40 might be due to adapting faulty rocket model from other site.

Payload vs. Launch Site



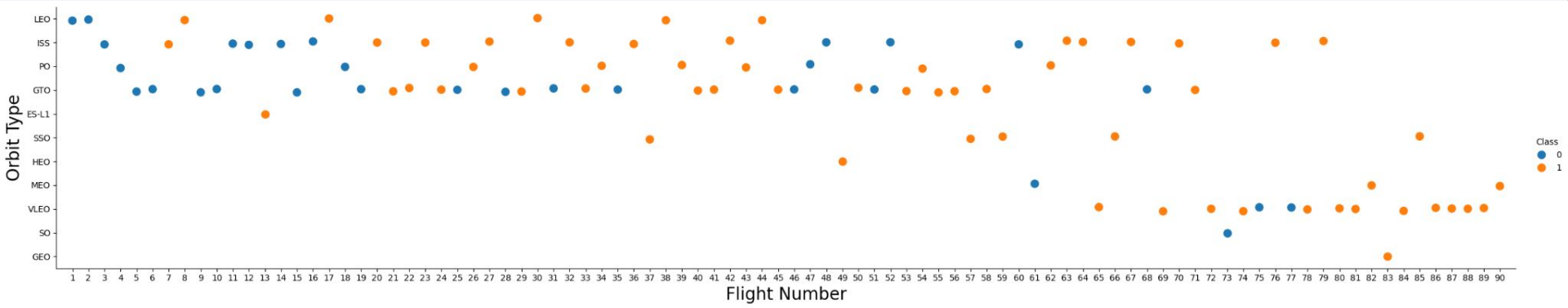
In VAFB, the bigger the payload, the more probability to get an successful mission outcome. Not exactly the same can be said for other two launch sites.

Success Rate vs. Orbit Type



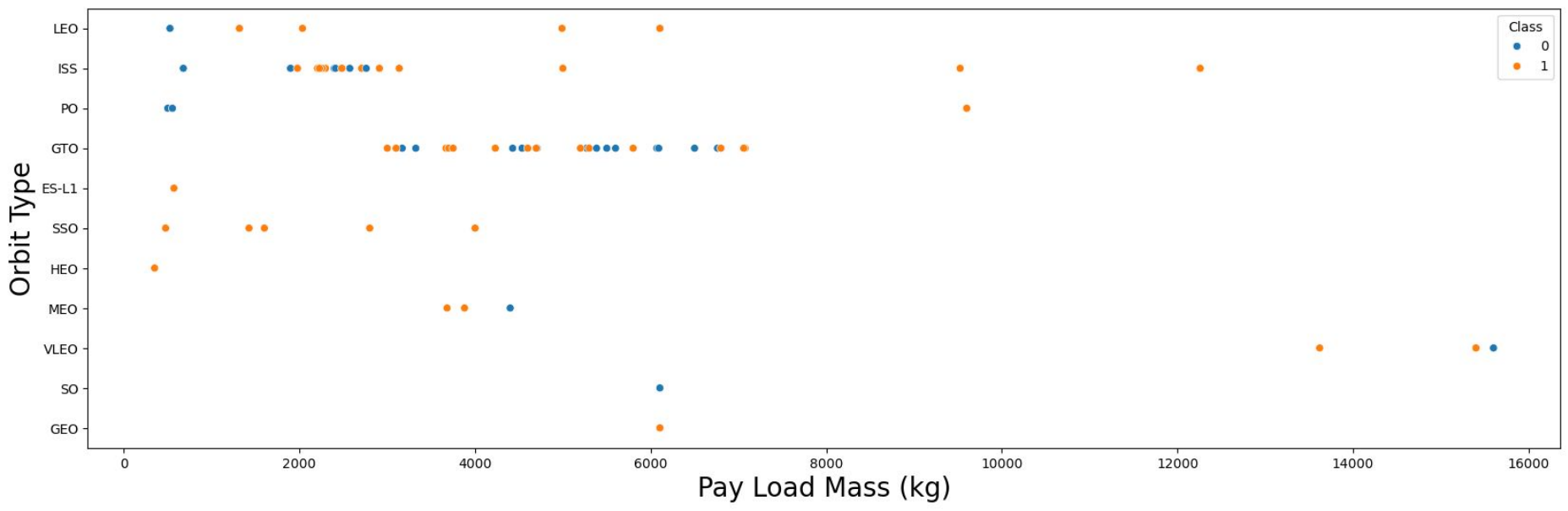
Looks like ES-L1, GEO, HEO and SSO have the largest success rate across orbits. It might be due to the probably more expensive equipment being sent there and thus requiring more precision and sureness for mission.

Flight Number vs. Orbit Type



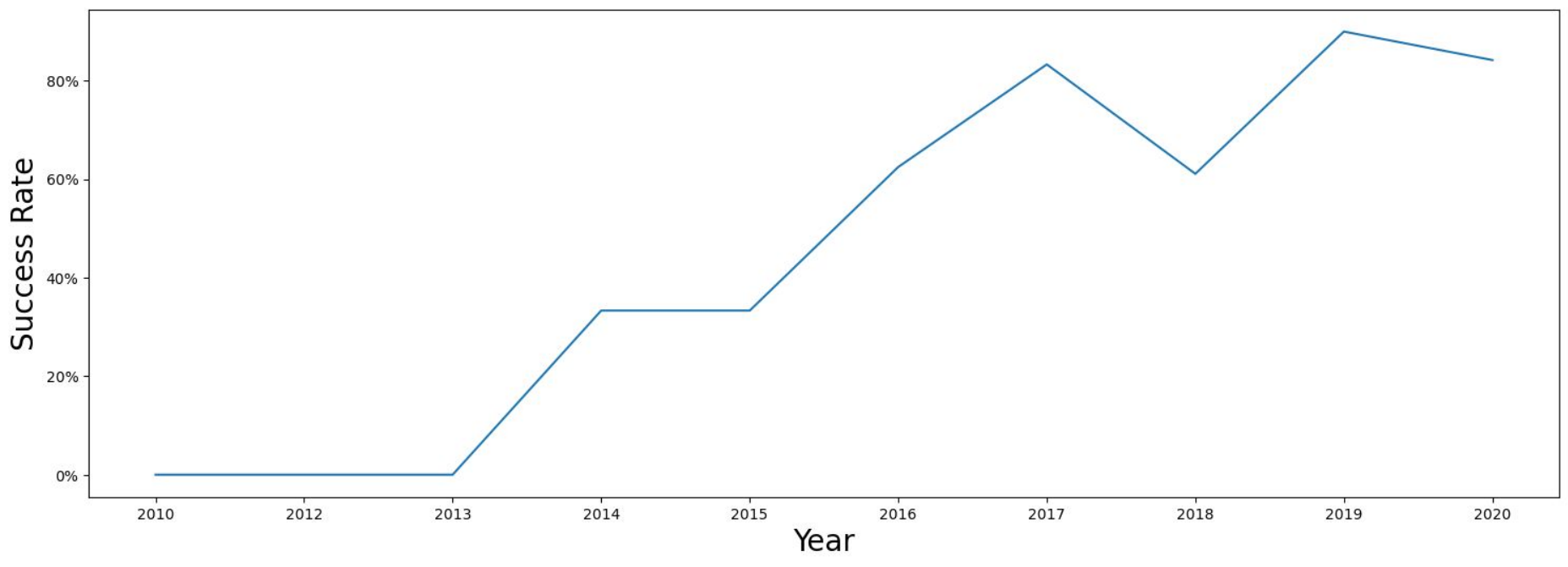
It can be observed that in a larger time scale of missions, there was also a shift in desired orbits. From the top 5 to the bottom ones on Y axis. Also the success rate from previous bar chart can now be explained in more detail while seeing, that some of the orbits were used really rarely and there are not enough samples to determine their true success rate or even having and impact on mission outcome.

Payload vs. Orbit Type



For some orbits (LEO, ISS, PO) the probability of successful outcome of a mission is positively correlated with payload mass being transported there.

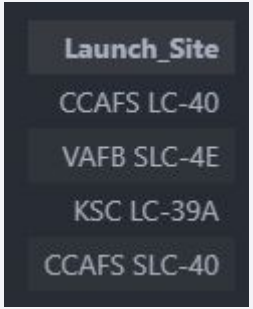
Launch Success Yearly Trend



While having a small dip in 2018, it can be observed that success rate increases across years, probably due to technology, funds and experience.

All Launch Site Names

```
%sql SELECT DISTINCT("Launch_Site") FROM SPACEXTABLE;
```



There are 4 different launch sites. Both of CCAFS are similarly named, because they are very close to each other.

Launch Site Names Begin with 'CCA'

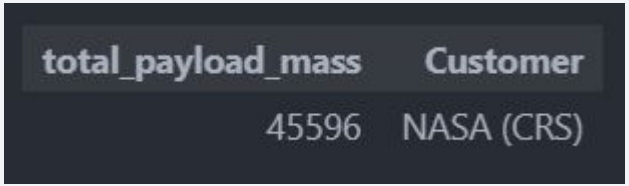
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

```
%%sql
SELECT * FROM SPACEXTABLE
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5
```

First 5 records from both CCAFC launch sites where between 2010 and 2013. All of them aiming for LEO orbit.

Total Payload Mass

```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload_mass, "Customer"
FROM SPACEXTABLE
WHERE "Customer" = 'NASA (CRS)'
```



total_payload_mass	Customer
45596	NASA (CRS)

Total payload mass used in rockets being serviced for NASA (CRS) was 45 596 kg.

Average Payload Mass by F9 v1.1

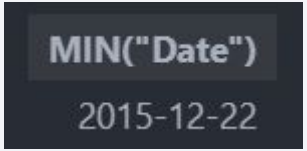
```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS average_payload_mass
FROM SPACEXTABLE
WHERE "Booster_Version" LIKE 'F9 v1.1%'
```

average_payload_mass
2534.6666666666665

Average payload mass being loaded into Falcon 9 booster v1.1 was about 2 535 kg.

First Successful Ground Landing Date

```
%%sql
SELECT MIN("Date")
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)'
```



First successful ground pad landing has taken place 2015-12-22.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT DISTINCT("Booster_Version")
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" BETWEEN
4001 AND 5999
```

FT B1022, FT B1026, FT B1021.2 and FT B1031.2 were the Falcon 9 boosters that had transported payload mass **greater than** 4000 kg but **less than** 6000 kg and successfully landed on a drone ship.

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT
   CASE
      WHEN "Mission_Outcome" LIKE '%success%' THEN 'success'
      ELSE 'failure'
   END AS outcome_case,COUNT(*) AS total_number_of_outcomes
FROM SPACEXTABLE
GROUP BY outcome_case
```

outcome_case	total_number_of_outcomes
failure	1
success	100

This data set is clearly unbalanced.

Boosters Carried Maximum Payload

```
%%sql
SELECT DISTINCT("Booster_Version")
FROM SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_")
FROM SPACEXTABLE)
```

These all of the booster version that carried the maximum payload mass from this data set. Looks like these iterations are after B1048.

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

```
%%sql
SELECT SUBSTR("Date", 6, 2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Failure (drone ship)' AND SUBSTR("Date", 0, 5) = '2015'
```

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

Both of the missions that failed to land on a drone ship in 2015 has taken place in the first half of the year.

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

```
%%sql
SELECT "Landing_Outcome", COUNT(*) AS num_of_outcomes
FROM SPACEXTABLE
GROUP BY "Landing_Outcome"
HAVING "Date" BETWEEN '2010-06-04' AND '2017-03-20'
ORDER BY num_of_outcomes DESC
```

Landing_Outcome	num_of_outcomes
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

The most common landing outcomes between 2010-06-04 and 2017-03-20 in order are: No attempt, Success (drone ship) and Success (ground pad)

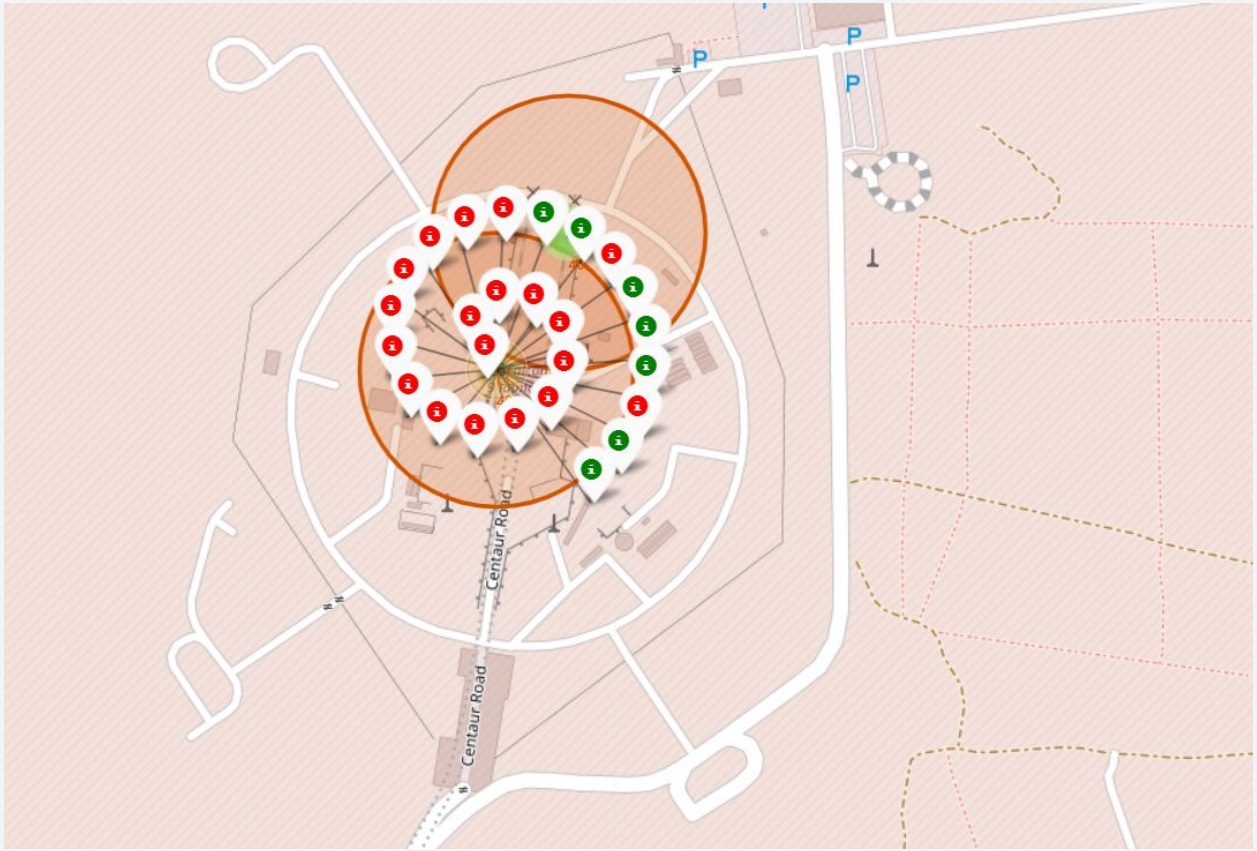
Section 3

Launch Sites Proximities Analysis

Map of all launch sites

It looks like all of the launch sites are close to the coastlines and close to the equator line. Not only it provides more safety against fires and dumping filled with explosive fuel rocket to the ocean, but closer proximity to equator helps to mitigate fuel usage.

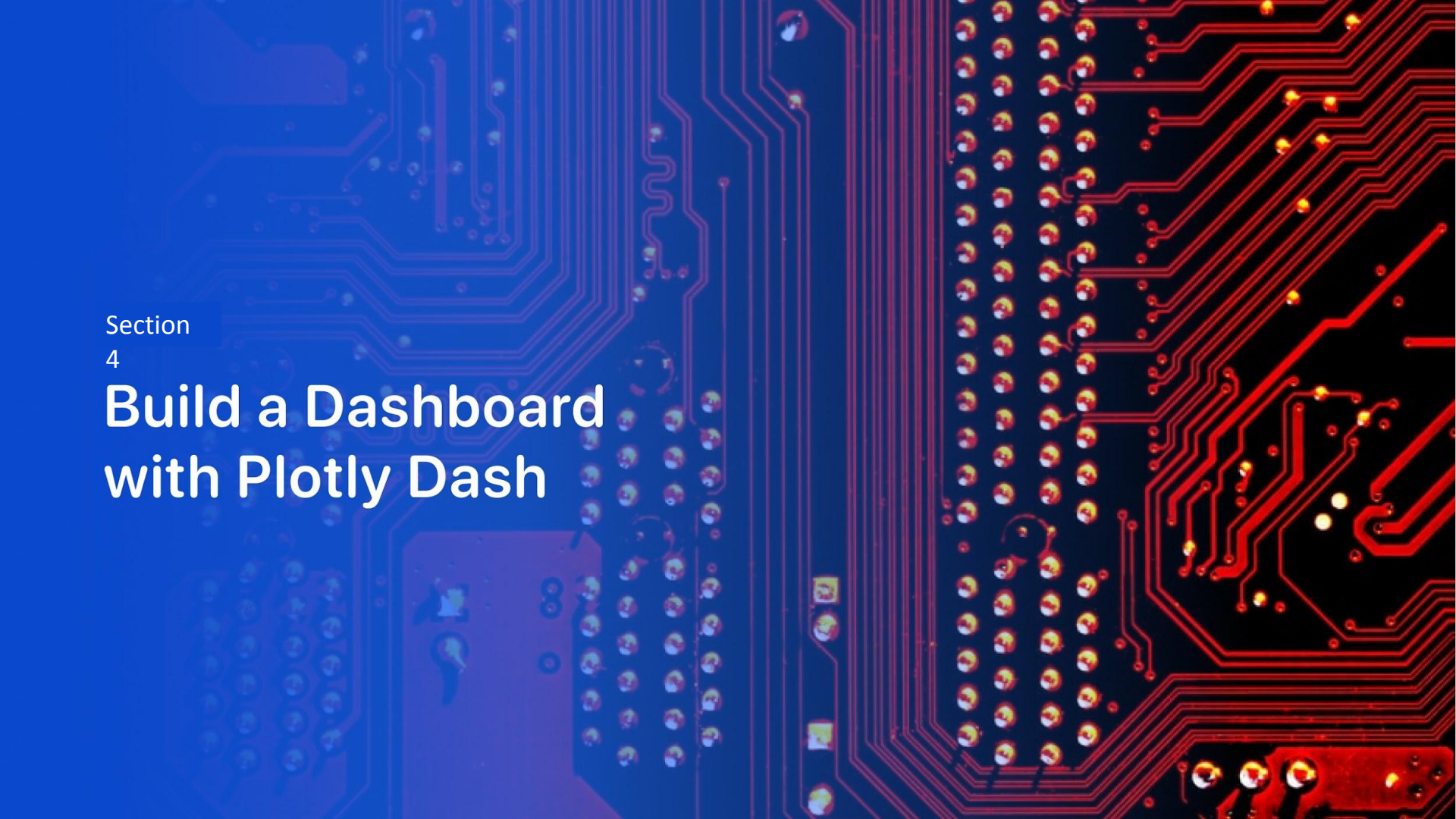
Map of colored outcomes for launch sites



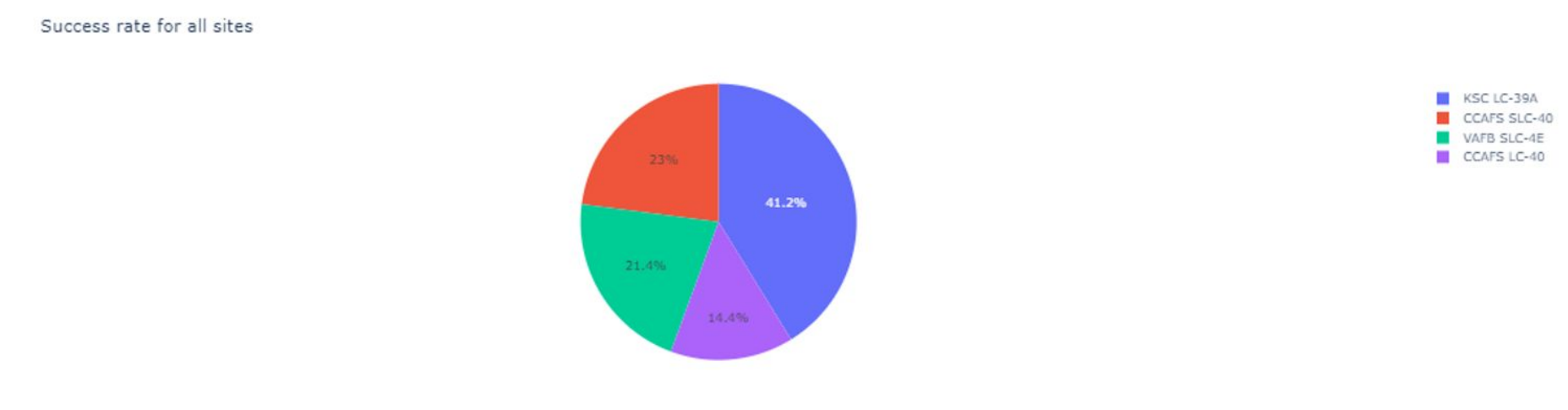
Viewer is able to deduct from the map how many successful and unsuccessful missions started from certain launch sites. The one shown here (CCAFS LC-40) seems to have lower success rate, with a lot of red (failure outcomes) indicators.

Proximity of landmarks

It seems that railroads, highways and coastlines are not so far away from this launch site, but city (not fully visible in this zoom) is much further away (18 km to Cape Canaveral), because of safety reasons.

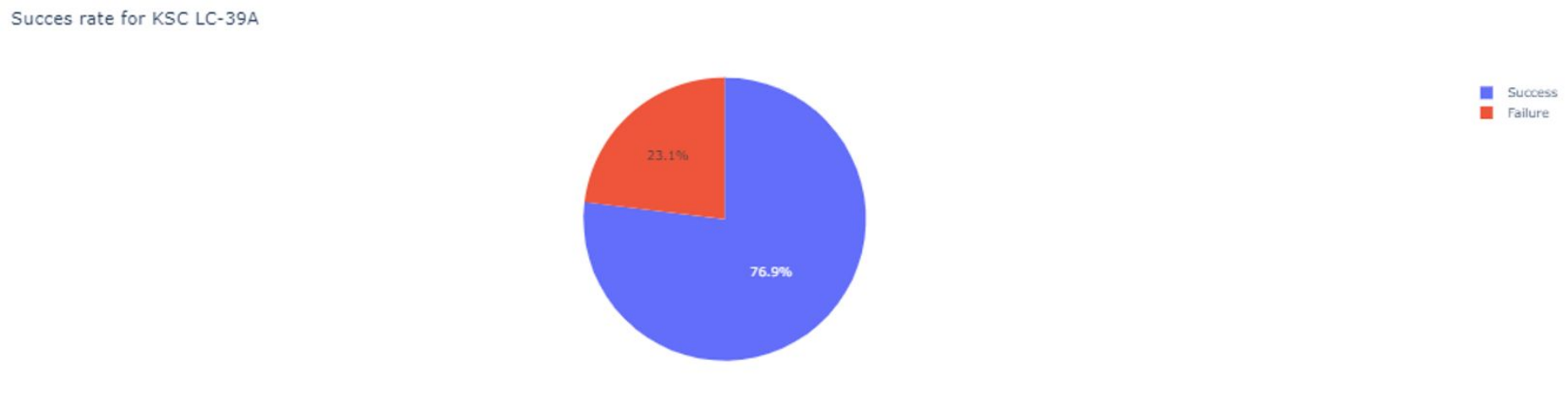

Section
4
Build a Dashboard
with Plotly Dash

Launch success count for all sites



It seems that KSC LC-39A is the launch site where there were the most successful missions.
On the opposite side, CCAFS LC-40 has the lower success rate of all launch sites.

Success rate for KSC LC-39A



One could deduct from this data set that for every 4 missions launched from KSC LC-39A, on average 3 would be successful.

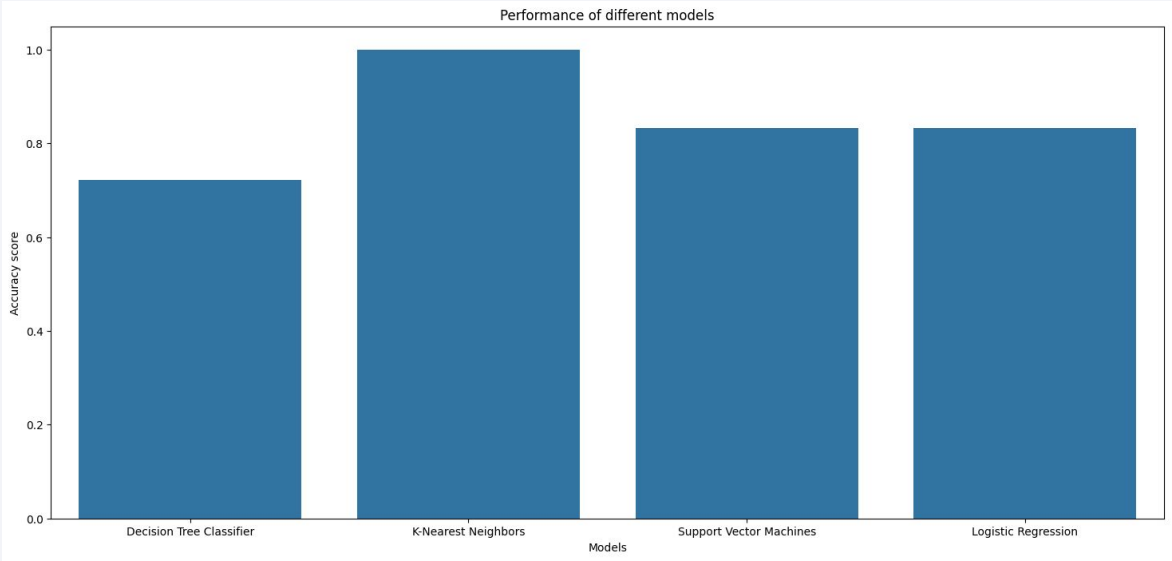
Interactive scatter plot

In the payload mass range between 3000 and 8000 kg, it is visible that successful launches usually are not that varied with payload and tend to be around 3500 and 5000 kg. Also v1.1 booster version category performed really poorly in this payload mass range.

Section
5

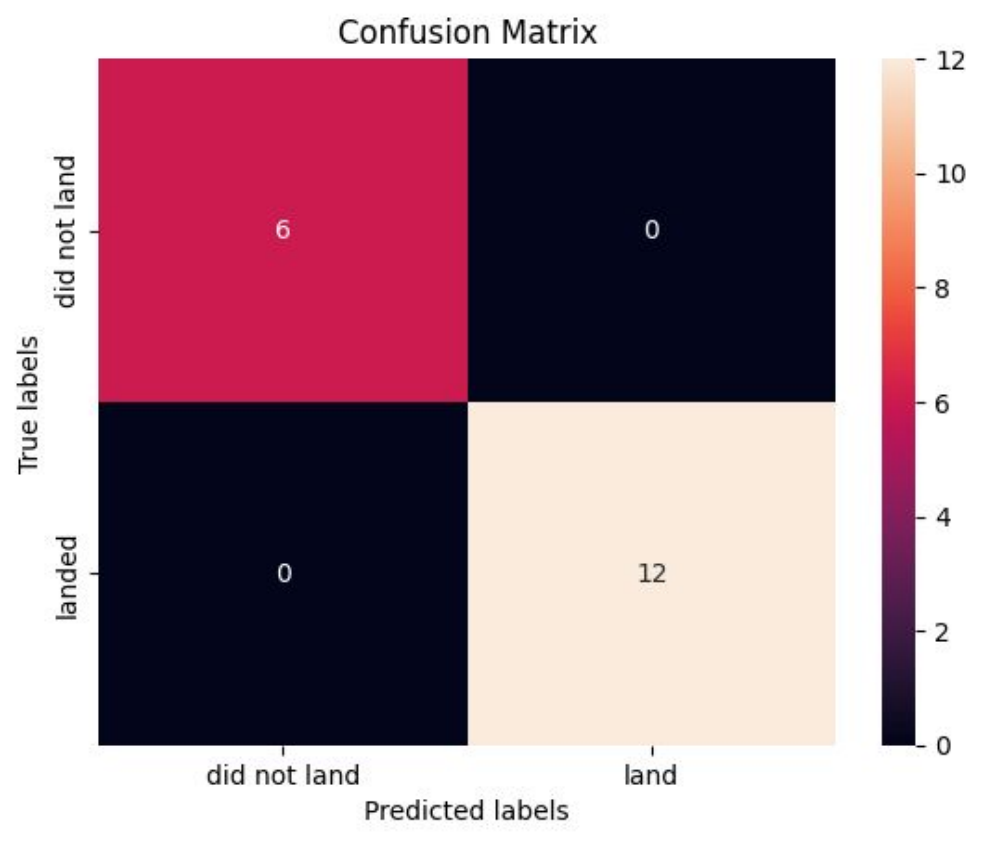
Predictive Analysis (Classification)

Classification Accuracy



K-Nearest Neighbors achieved the highest accuracy score on a test set.

Confusion Matrix of KNN



Confusion Matrix of best performing model (KNN) shows that every label was predicted correctly, with zero false positives and zero false negatives.

Conclusions

- Despite having small data set, it was possible to predict mission outcome from features, KNN algorithm achieves 100% accuracy - but only once, on predefined test set.
- Most of models performed good on this data, which indicates that features explain really well the mission outcome.
- SpaceY can not only analyze specifics of reusable boosters from different visualizations and interactive dashboards/maps, but maybe gain knowledge of what solutions perform best in this area.
- Solutions of SpaceX can fuel future technology of rocket engineering, all thanks to their open-source character of data, broadcasts.
- Reusability of rockets proved in this project not only its superiority over not reusable rockets, but also some degree of predictability, which lowers risk greatly for companies interested in investing into a space race.

Potential improvements

- Subject is hard to evaluate accurately, because every row of data costs about 80 million $. This is why the data set used for this project had only 90 rows.
- Data is unbalanced, there are twice as much successful records then unsuccessful. Better method to tackle this problem would be to use stratified version train_test_split.
- Grid search with 10 folds is not a best idea for parsing through only 72 records of training set after splitting. This way model is cross-validated on only 7 samples - high probability for old unsuccessful records or for imbalance set. It would be better to optimize hyperparameters with optuna or other modules and create custom study that would cross-validate on stratified folds.
- KNN model achieved the best performance, but its k value was set to 1, which is prone to overfitting. This model performed good on this data set, but can be a lot worse on other data.
- Machine explainability could be an another chapter. Thanks to use of mutual information, permutation importance, partial dependence plot and Shapely values, one can extract more information from this data to determine which features affect the mission outcomes the most and in what way.

Appendix

Thank you!