Coursera Data Science Capstone Project:

# Exploratory Data Analysis and Prediction of Successful Landings for Space X's Space Rockets First Stage

**Andrei Karavai**

**13. August 2021**

# Contents

# Executive Summary

- This project aims to predict if the Falcon 9 Space Rocket first stage will land successfully and to provide EDA concerning Space X rocket launches and firs stage landings.
- In this project data on Space X rocket launches was gathered from internet (web-scraping, API requests). It was further processed with tools available in Python programming language. EDA was provided with help of various visualization tools (including interactive) and with help of SQL quires. Several classification models (Logistic Regression, SVM, Decision Tree and KNN) were trained and optimized to predict the success of first stage landing. Best performing model was chosen with help of accuracy score. Prediction quality was analyzed with help of confusion matrix
- EDA shows that a list of different factors affect the success of first stage landing. High accuracy score (over 80%) could be achieved for predictions based on gathered datasets. SVM method provides best accuracy score result (Test Set Score: 0,83; Entire Dataset Score: 0,88). False positives rate is the point for further improvement of the predicting model based on confusion matrix analysis.

# Introduction

## Project background and context

Space X company offers Falcon 9 rocket launches on its website with a cost of 62 million USD whereas other providers offer launches at cost of 165 million USD. Major part of the savings is because Space X can reuse the first stage. If other company can determine if the first stage will successfully land, it can predict the cost of a launch. This information can be used to bid against space X for a rocket launch.

## Tasks for project

There are basic task and additional (optional) tasks set for this project:

**Basic task:** Predict if the Falcon 9 Space Rocket first stage will land successfully.

**Additional (optional) tasks:**

- Provide exploratory data analysis concerning Space X rocket launches and firs stage landings.
- Gather insights on conditions of successful and unsuccessful landings.

# Methodology (Overview)

## Data collection methodology:

For this project, the relevant data was:

- Requested from the SpaceX REST API endpoints (https://api.spacexdata.com/v4).
- Scraped from Wikipedia Web-page (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)

## Data wrangling:

For this project data wrangling consisted of:

- Removing not relevant data-records.
- Replacement of missing values with average values.
- Exploration of data-types in the given data-sets.
- Setting of Y-variable (Label) for further model training.

## Exploratory data analysis (EDA) using visualization and SQL:

- The dependencies between features of datasets were visualized and explored with charts and plots.
- Additional insights in the provided datasets were made with help of SQL queries.
- Features for prediction (independent variables) were prepared based on the results of visual analysis.

## Interactive visual analytics using Folium and Plotly Dash:

- Additional analysis of geographical patterns of launch sites was made with help of Folium package.
- Interactive data visualization with help of Python Plotly Dash package was prepared to get better insight in provided data.

## Predictive analysis using classification models:

For the predictive analysis:

- Features (independent variables) were standardized with .StandardScaler() function.
- Dependent and independent variables set were split into test set and training set for model training.
- Logistic Regression, SVM, Decision tree and KNN methods were used for predictive models.
- Best Hyperparameters for models were chosen with GridSearchCV.
- The predictive quality of models was compared using confusion matrix and .score() function.

# Methodology

# Data collection (1/3)

For this project, the relevant data was:

- Requested from the SpaceX REST API endpoints (https://api.spacexdata.com/v4).
- Scraped from Wikipedia Web-page (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)

Data collection process is presented on the following 2 slides.

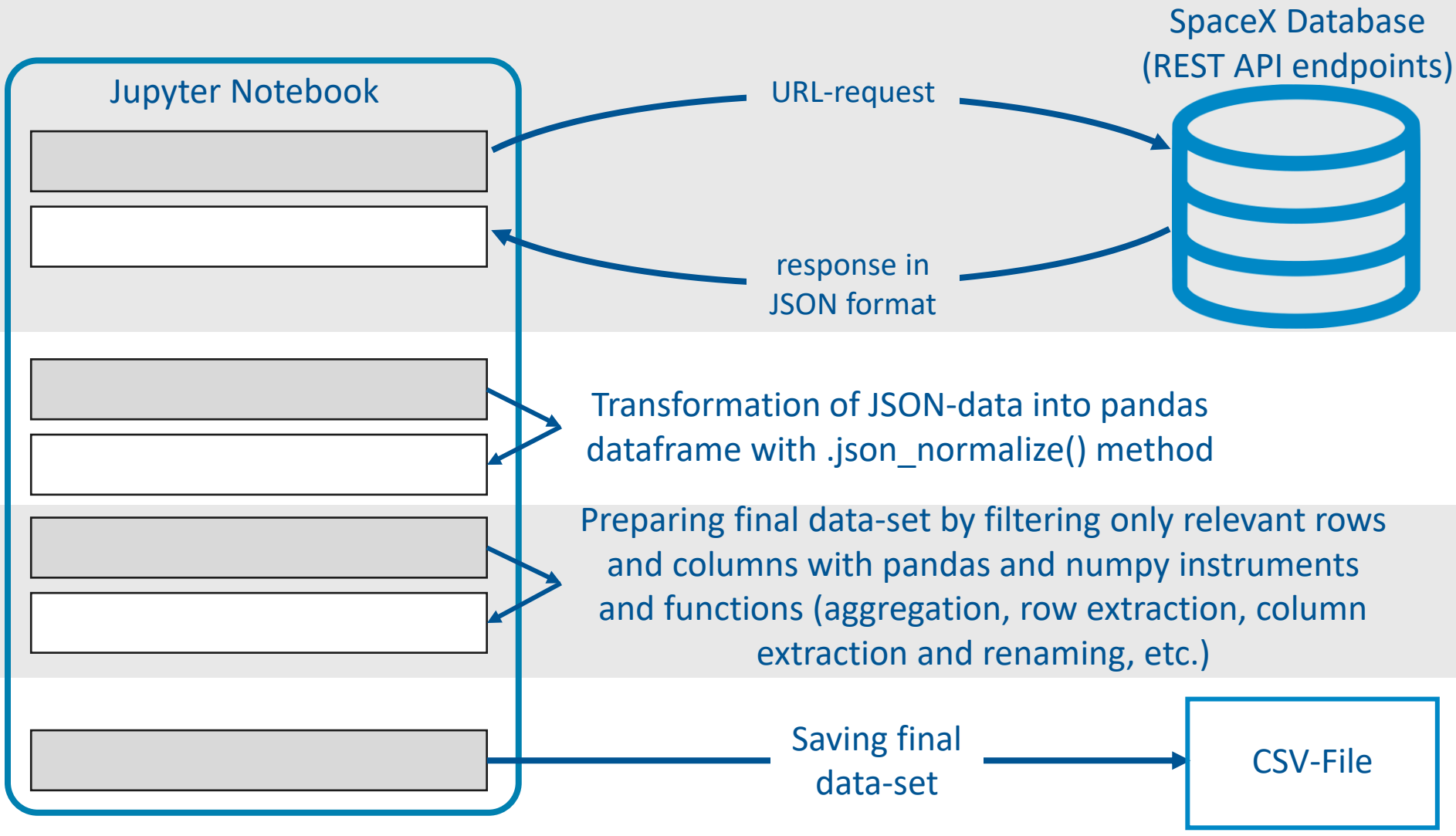
# Data collection (2/3): SpaceX API calls and data processing scheme

1. Getting data from Space X API endpoints

2. Transforming JSON into pandas dataframe

3. Preparing final data-set

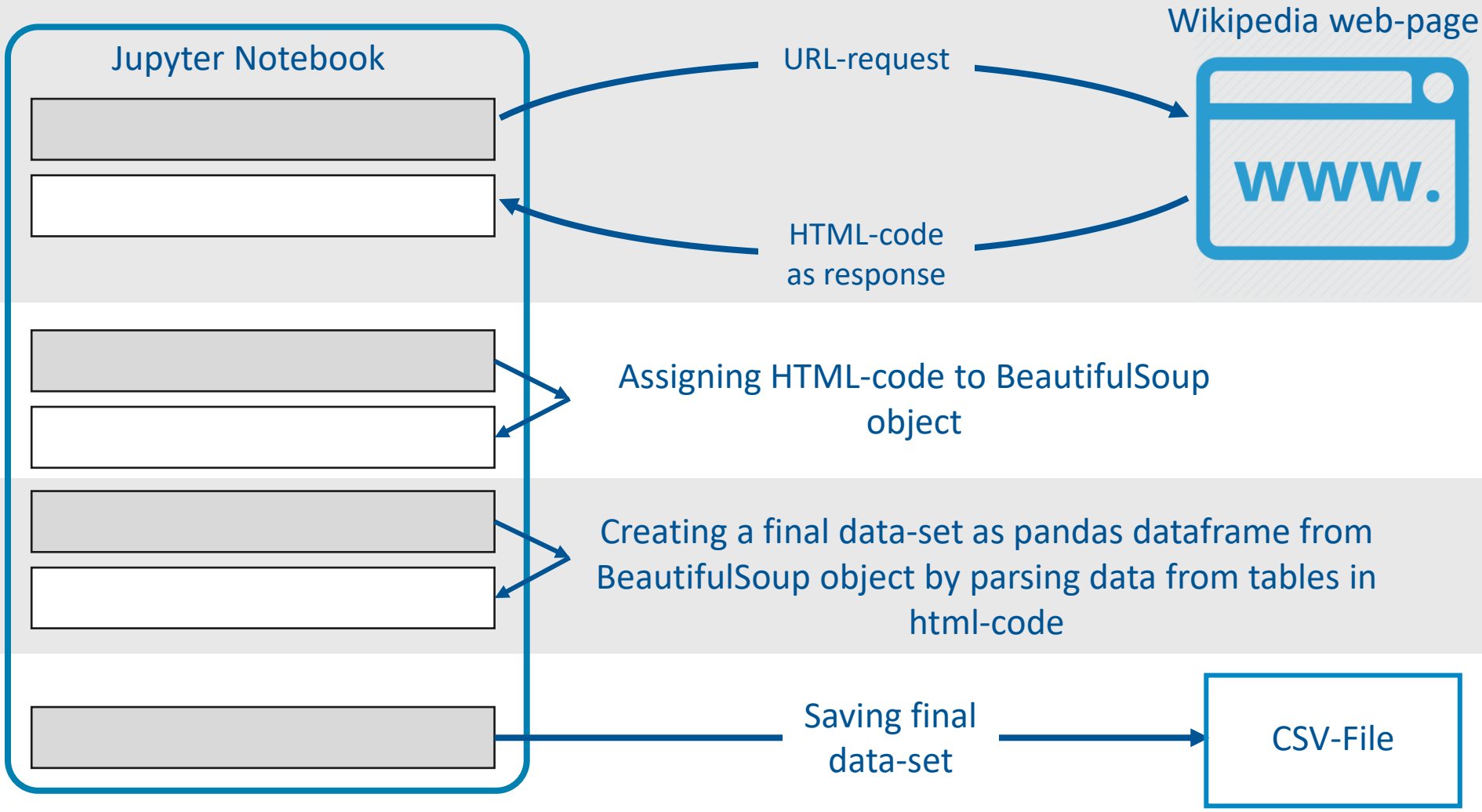
4. Saving final data-set to CSV-File




**GitHub URL of the completed SPACE X API notebook:**
https://github.com/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK1_jupyter-labs-spacex-data-collection-api.ipynb
https://nbviewer.jupyter.org/github/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK1_jupyter-labs-spacex-data-collection-api.ipynb

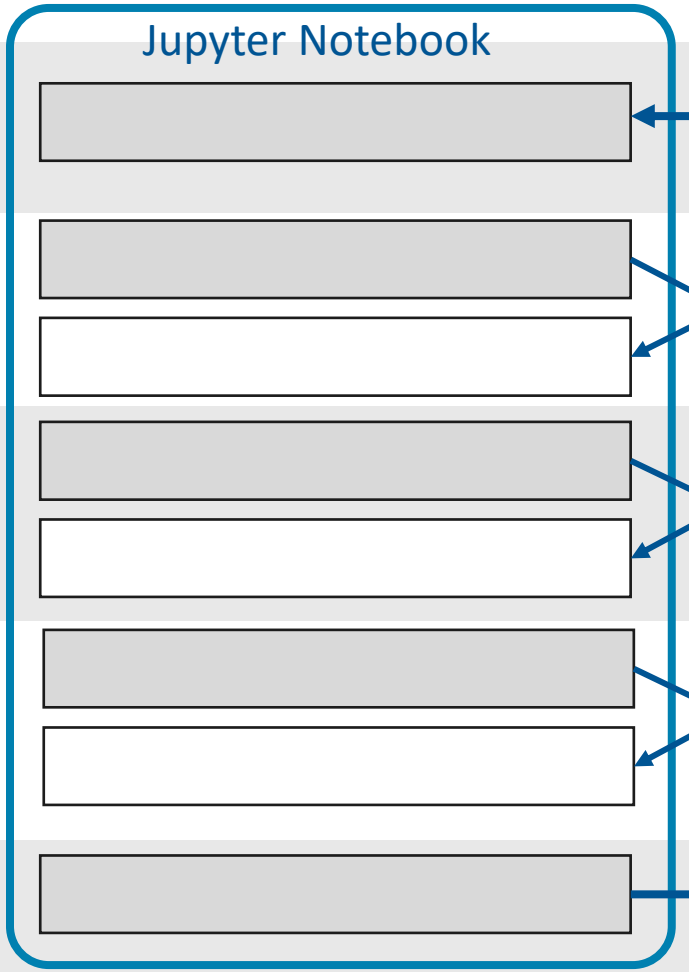
# Data collection (3/3): Scheme of web-scraping form Wikipedia

1. Getting data from Wikipedia
2. Creating Beautiful Soup object from HTML-code
3. Preparing final data-set
4. Saving final data-set to CSV-File




**GitHub URL of the completed web scraping notebook:**
https://github.com/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK1_jupyter-labs-webscraping.ipynb
https://nbviewer.jupyter.org/github/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK1_jupyter-labs-webscraping.ipynb

# Data wrangling




1. Loading data-set from a CSV-file

Using pandas.read_csv()

CSV-File

2. Exploring data in each column and looking for missing values

Using dataframe.dtypes(), dataframe.value_counts() and dataframe.isnull()

3. Replacing missing values of payload mass

Using dataframe.replace() and dataframe.mean().
(This part is done in Task 3 of Space X API calls lab)

4. Creating label (Y-variable) for model training

Creating new column 'CLASS' and filling it with 0s and 1s depending on first stage landing status in column 'Outcome'

5. Saving final data-set to CSV-File

Using dataframe.to_csv()

CSV-File

**GitHub URL of the completed data wrangling related notebooks:**
https://github.com/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK1_labs-jupyter-spacex-Data%20wrangling.ipynb
https://nbviewer.jupyter.org/github/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK1_labs-jupyter-spacex-Data%20wrangling.ipynb

https://github.com/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK1_jupyter-labs-spacex-data-collection-api.ipynb (Point 3)
https://nbviewer.jupyter.org/github/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK1_jupyter-labs-spacex-data-collection-api.ipynb (Point 3)

# EDA with data visualization

| Name of Chart | Type of Chart | Purpose of Chart |
|---|---|---|
| Landing outcome for Flight Number vs. Payload Mass | Scatter | To check if landing success rate increase with for later flights and to see if success rate is higher for higher payload mass. |
| Landing outcome for Flight Number vs. Launch Site | Scatter | To check the distribution of launches between the launch sites in time. To check the change of success rate from earlier launches to later launches for each launch site. |
| Landing outcome for Payload Mass vs. Launch Site | Scatter | To check the distribution of launches with different payload mass between the launch sites. To check the payload mass range that has high and low success rates. |
| Success rate for Orbit Type | Bar | To see if different orbits have different success rates. |
| Landing outcome for Flight Number vs. Orbit Type | Scatter | To check the change of success rate from earlier launches to later launches for each orbit type. |
| Landing outcome for Payload Mass vs. Orbit Type | Scatter | To check the influence of payload mass on success rate for each orbit type. |
| Launch Success Yearly Trend | Line | To check the change of success rate from 2013 till 2020 |

See charts on slides 17-23

**GitHub URL of the completed EDA with data visualization notebook:**
https://github.com/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK2_jupyter-labs-eda-dataviz.ipynb
https://nbviewer.jupyter.org/github/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK2_jupyter-labs-eda-dataviz.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 with help of subquery
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
- Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order

See the queries and results on slides 24-34

**GitHub URL of the completed EDA with SQL notebook:**
https://github.com/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK2_jupyter-labs-eda-sql-coursera.ipynb
https://nbviewer.jupyter.org/github/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK2_jupyter-labs-eda-sql-coursera.ipynb

# Interactive map with Folium

| Map Objects Added to Map | Type of Map Object | Purpose of Object |
|---|---|---|
| NASA Johnson Space Center Markers | Circle, Popup Label, Text Label | To show the location of NASA command centre on the map. |
| Markers for Every Launch Site | Circle, Popup Label, Text Label | To show the location of every launch site on the map. |
| Markers of success/failed launches for each launch site | Color-Labeled Marker | To identify success rate for every launch site. |
| Lines to show distance to nearest railway, city, coast, highway | Polyline, Text Label | To measure the distance to the nearest railway, highway, city, coast. |

See the Map screenshots on slides 35-38

**GitHub URL of the completed interactive map with Folium notebook:**
https://github.com/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK3_lab_jupyter_launch_site_location.ipynb
https://nbviewer.jupyter.org/github/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK3_lab_jupyter_launch_site_location.ipynb **<--To see rendered maps use this link!**

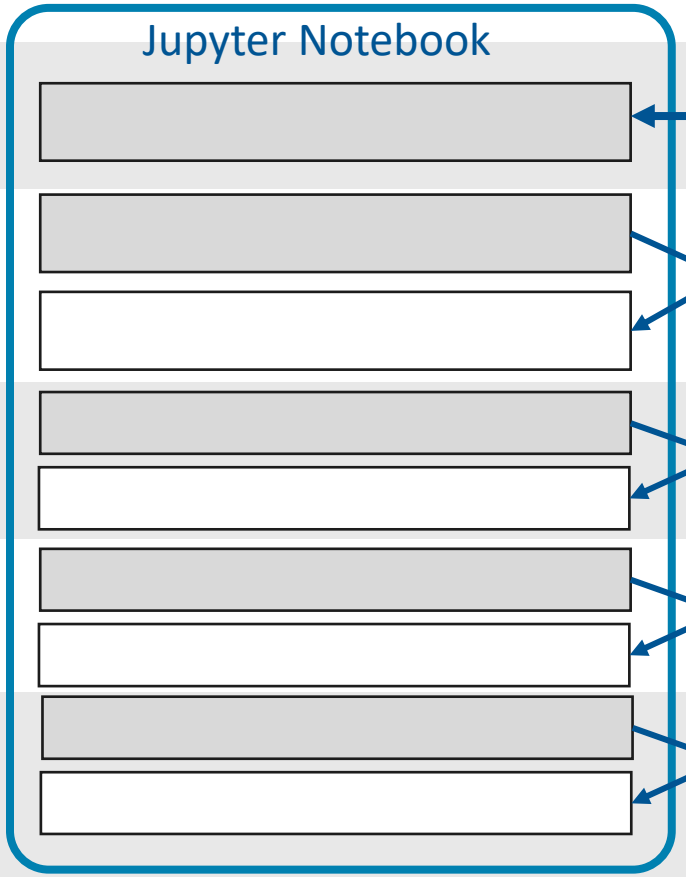
# Build a Dashboard with Plotly Dash

| Created Dashboard objects | Purpose of Object |
|---|---|
| Launch Site Dropdown List | To enable interactive Launch Site selection for charts |
| Pie Chart of Successful Launches | Shows the total successful launches count for all sites if all Launch Sites are selected. If a specific launch site is selected, the pie chart shows the Success vs. Failed counts for the site. |
| Slider of Payload Mass Range | To select the range of payload mass for charts where payload mass is used |
| Scatter Chart: Booster version for Success Rate vs. Payload Mass | To show the correlation between payload and launch success for each booster version |

See the Dashboard screenshots on slides 39-42

**GitHub URL of the completed Plotly Dash lab:**
https://github.com/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK3_Plotty_dash.ipynb **<--Plotly Dash lab in Jupyter Notebook**
https://github.com/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK3_spacex_dash_app.py **<--Plotly Dash lab in Python (*.py) file**

# Predictive analysis (Classification) workflow




1. Loading data-set from a CSV-file

Using pandas.read_csv()

CSV-File

2. Creating NumPy array with Y variable and standardizing data for X

Using dataframe. to_numpy(), preprocessing.StandardScaler().fit().transform()

3. Splitting data into train and test sets

Using train_test_split()

4. Training models with best parameters for each prediction method

Used methods: logistic regression, SVM, K nearest neighbors, decision tree. GridSearchCV() is used to find best parameters for each method.

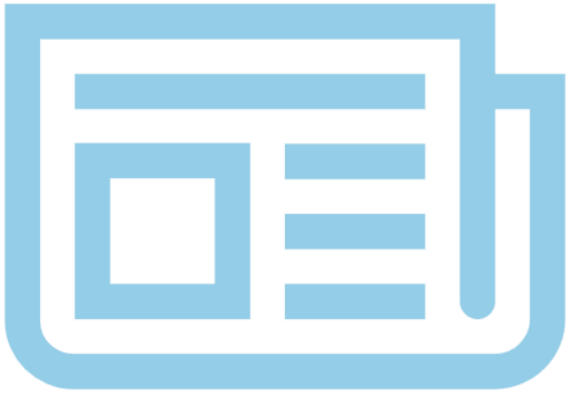
5. Evaluating the performance of each method and comparing the prediction results

Metrics to evaluate performance of the models: confusion matrix plot, accuracy score on test set, accuracy score on the entire data-set

See the details on predictive data analysis on slides 43-45

**GitHub URL of the completed predictive analysis lab:**
https://github.com/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK4_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb
https://nbviewer.jupyter.org/github/andrei-karavai/Coursera_Capstone2021/blob/main/WEEK4_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

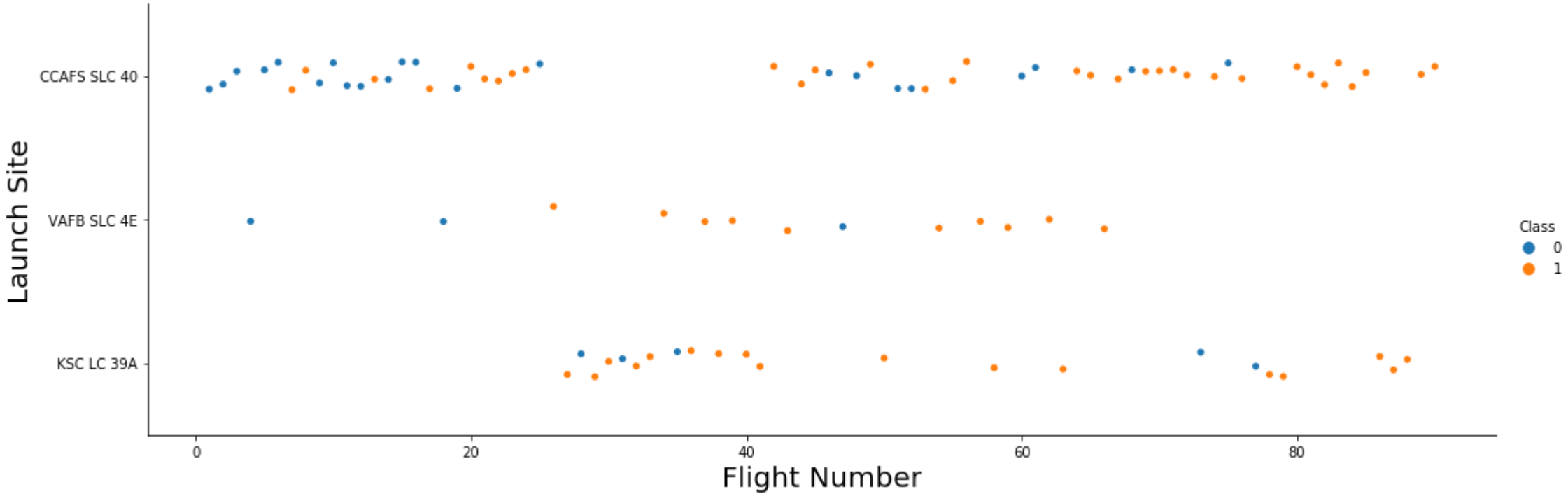
# Results



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

# EDA with Visualization

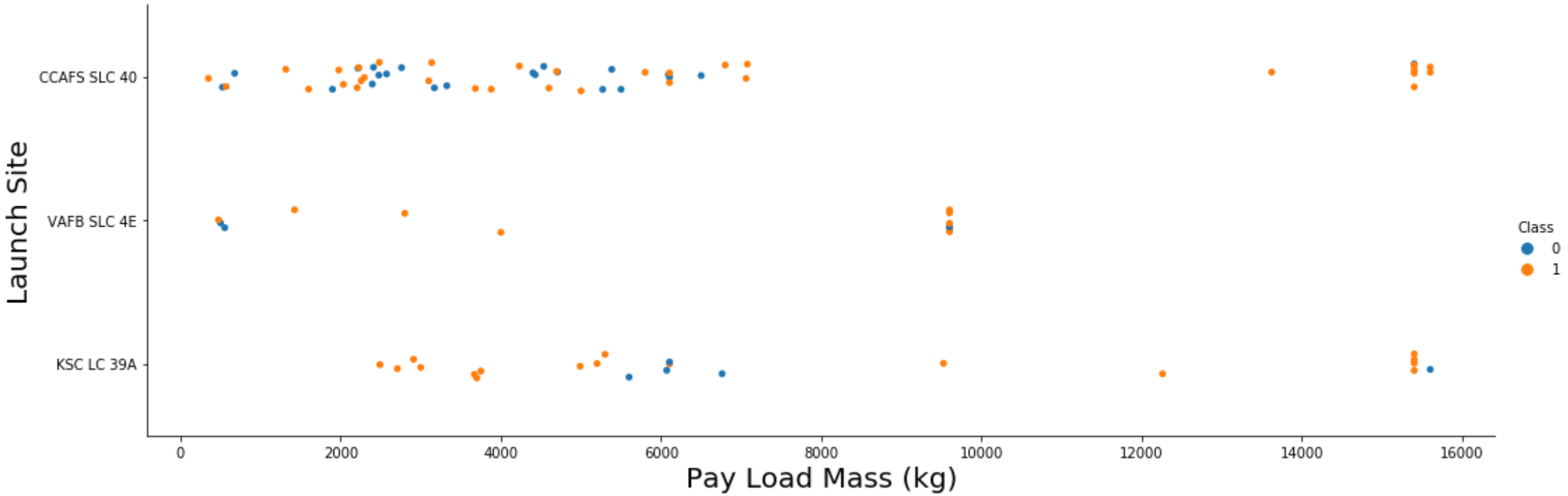
# Flight Number vs. Launch Site




Observations and conclusions:

- CCAFS SLC 40 launch site is used the most time.¶
- CCAFS SLC 40 launch site has the highest number of failed Launches at the beginning.
- Starting with Flight Number 78 all launches on all launch sites were successful.

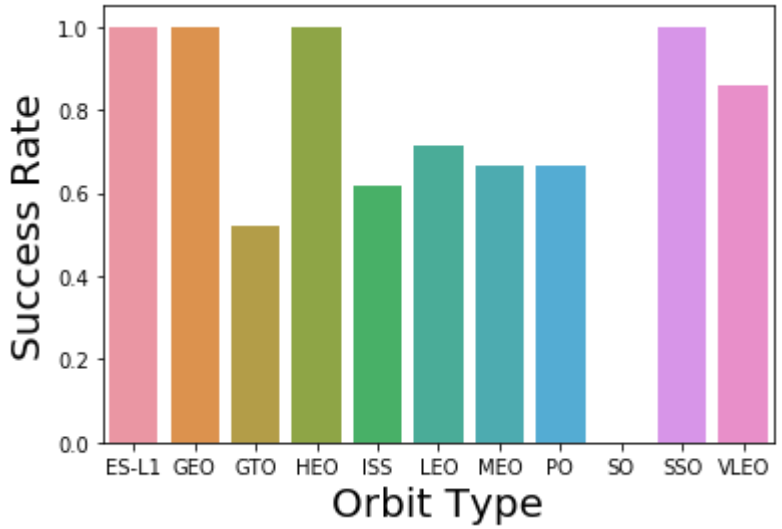
# Payload vs. Launch Site




Observations and conclusions:

- For every launch site the higher the payload mass is the higher is the success rate.
- KSC LC 39A launch site has the highest general success rate, but it has problems for payload mass in range from 5000 to 7000 kg.
- Most of unsuccessful launches had payload mass under 7000 kg.

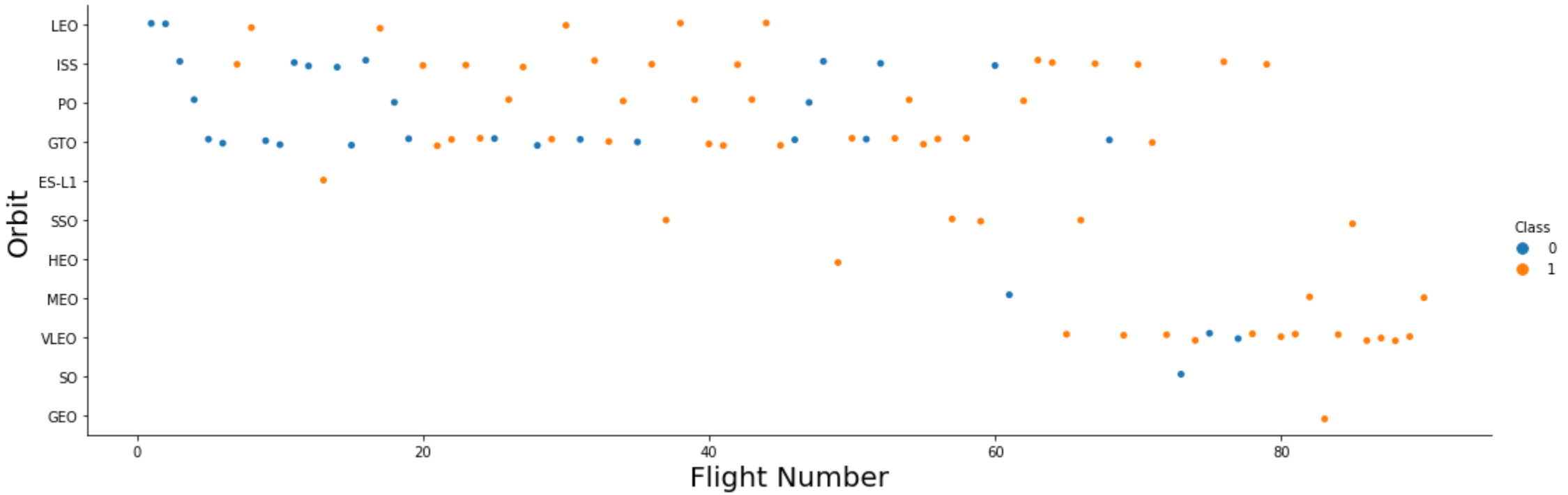
# Success rate vs. Orbit type




Observations and conclusions:

- ES-L1, GEO, HEO and SSO have 100% success rate.
- SO has 0% sucess rate.
- VLEO has sucess rate above 80%.
- GTO, ISS, LEO, MEO, PO have sucess rate in range from 50% to 80%.

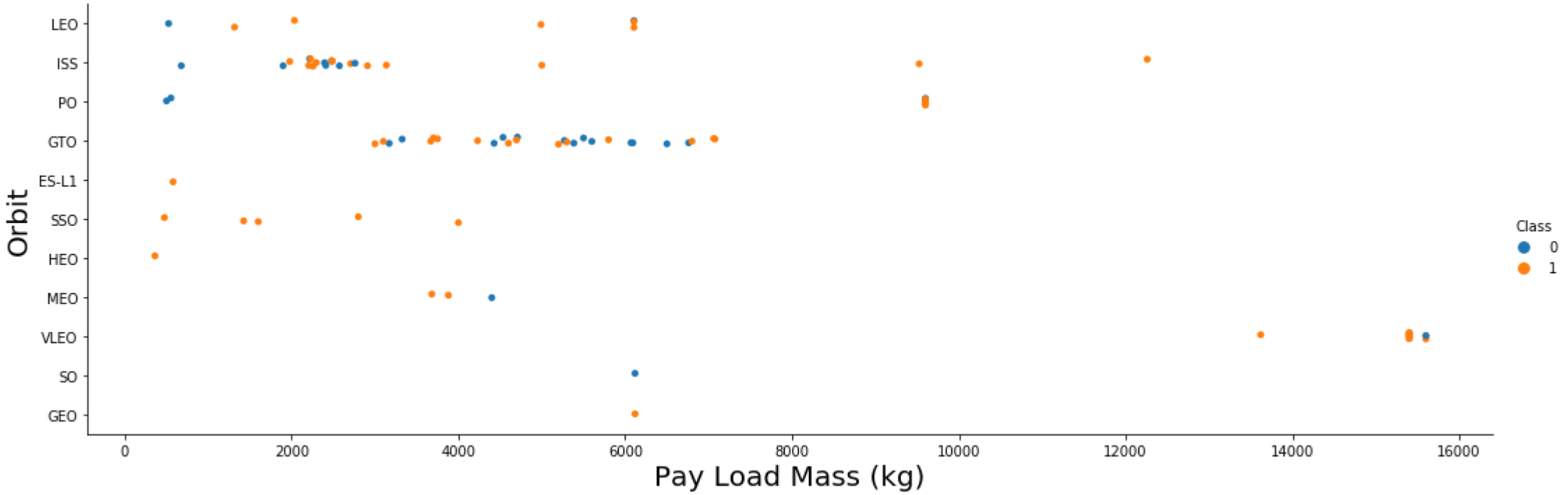
# Flight Number vs. Orbit type




Observations and conclusions:

- In the LEO orbit the Success is related to the number of flights.
- No relationship between flight number for GTO orbit.
- Launches to VLEO orbit were performed late: after flight #60

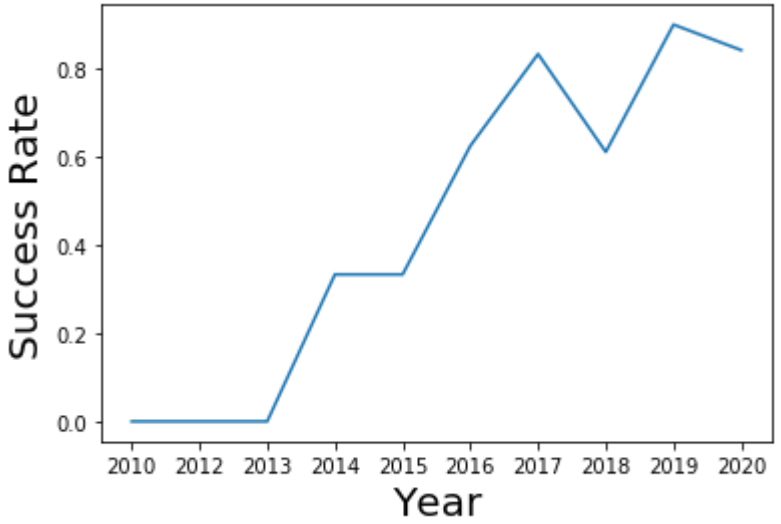
# Payload vs. Orbit type




Observations and conclusions:

- Heavy payloads have a negative influence on GTO orbits.
- Heavy payloads have a positive influence on LEO and ISS (Polar LEO) orbits.

# Launch success yearly trend




Observations and conclusions:

- The sucess rate since 2013 kept increasing till 2020.

# EDA with SQL

# All launch site names

- Find the names of the unique launch sites

In [4]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXDATASET
```

```
 * ibm_db_sa://vql23019:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[4]:

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

- 5 unique launch sites are used for Space X launches

# Launch site names begin with `CCA`

- Find all launch sites begin with `CCA`

In [14]:
```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXDATASET
WHERE LAUNCH_SITE LIKE 'CCA%'
```

```
 * ibm_db_sa://vql23019:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[14]:

| launch_site |
|---|
| CCAFS LC-40 |
| CCAFS SLC-40 |
| CCAFSSLC-40 |

- 3 unique launch sites have names starting with ‘CCA’

# Total payload mass

- Calculate the total payload carried by boosters from NASA

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

```
In [6]: %sql SELECT SUM(payload_mass__kg_) FROM SPACEXDATASET WHERE CUSTOMER='NASA (CRS)'
```

```
 * ibm_db_sa://vql23019:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[6]:

| 1 |
|---|
| 45596 |

- Total payload carried by boosters launched by NASA (CRS) is 45 596 kilogram

# Average payload mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

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

```
In [7]: %sql SELECT AVG(payload_mass__kg_) FROM SPACEXDATASET WHERE BOOSTER_VERSION LIKE 'F9 v1.1%'
```

```
 * ibm_db_sa://vql23019:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[7]:

| 1 |
|---|
| 2534 |

- Average payload mass carried by F9 v1.1 is 2 534 kilogram

# First successful ground landing date

- Find the date when the first successful landing outcome in ground pad

***List the date when the first succesful landing outcome in ground pad was acheived.***

*Hint:Use min function*

```
In [8]: %sql SELECT MIN(DATE) FROM SPACEXDATASET WHERE LANDING__OUTCOME='Success (ground pad)'
 * ibm_db_sa://vql23019:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
Out[8]:      1
          2015-12-22
```

- First successful landing on ground pad was achieved on the 22nd of December 2015

# Successful drone ship landing with payload between 4000 and 6000

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

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

In [15]:
```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXDATASET
WHERE LANDING__OUTCOME='Success (drone ship)' AND payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000
```

```
 * ibm_db_sa://vql23019:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[15]:

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

- There are 4 boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

# Total number of successful and failure mission outcomes

- Calculate the total number of successful and failure mission outcomes

***List the total number of successful and failure mission outcomes***

In [16]:
```
%%sql
SELECT MISSION_OUTCOME, Count(*) AS COUNT
FROM SPACEXDATASET
GROUP BY MISSION_OUTCOME
```

```
 * ibm_db_sa://vql23019:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[16]:

| mission_outcome | COUNT |
|---|---|
| Failure (in flight) | 1 |
| Success | 99 |
| Success (payload status unclear) | 1 |

- There are 100 Successful outcomes and 1 failure mission outcome.

# **Boosters carried** maximum **payload**

- List the names of the booster which have carried the maximum payload mass

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

In [11]:
```
%%sql
SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXDATASET
WHERE payload_mass__kg_=(SELECT MAX(payload_mass__kg_) FROM SPACEXDATASET)
```

```
 * ibm_db_sa://vql23019:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[11]:

| 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 |

- 12 boosters carried maximum payload mass

# 2015 launch records

- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015

***List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015***

In [12]:
```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXDATASET
WHERE EXTRACT(YEAR FROM DATE)=2015 AND LANDING__OUTCOME='Failure (drone ship)'
```

```
 * ibm_db_sa://vql23019:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[12]:

| MONTH | landing__outcome | booster_version | launch_site |
|---|---|---|---|
| January | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| April | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

- Total of 2 records, one for January and one for April

# Rank success count between 2010-06-04 and 2017-03-20

- Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

***Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.***

In [13]:
```
%%sql
SELECT LANDING__OUTCOME, COUNT(*) AS OUTCOME_COUNT
FROM SPACEXDATASET
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY OUTCOME_COUNT DESC
```

```
 * ibm_db_sa://vql23019:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[13]:

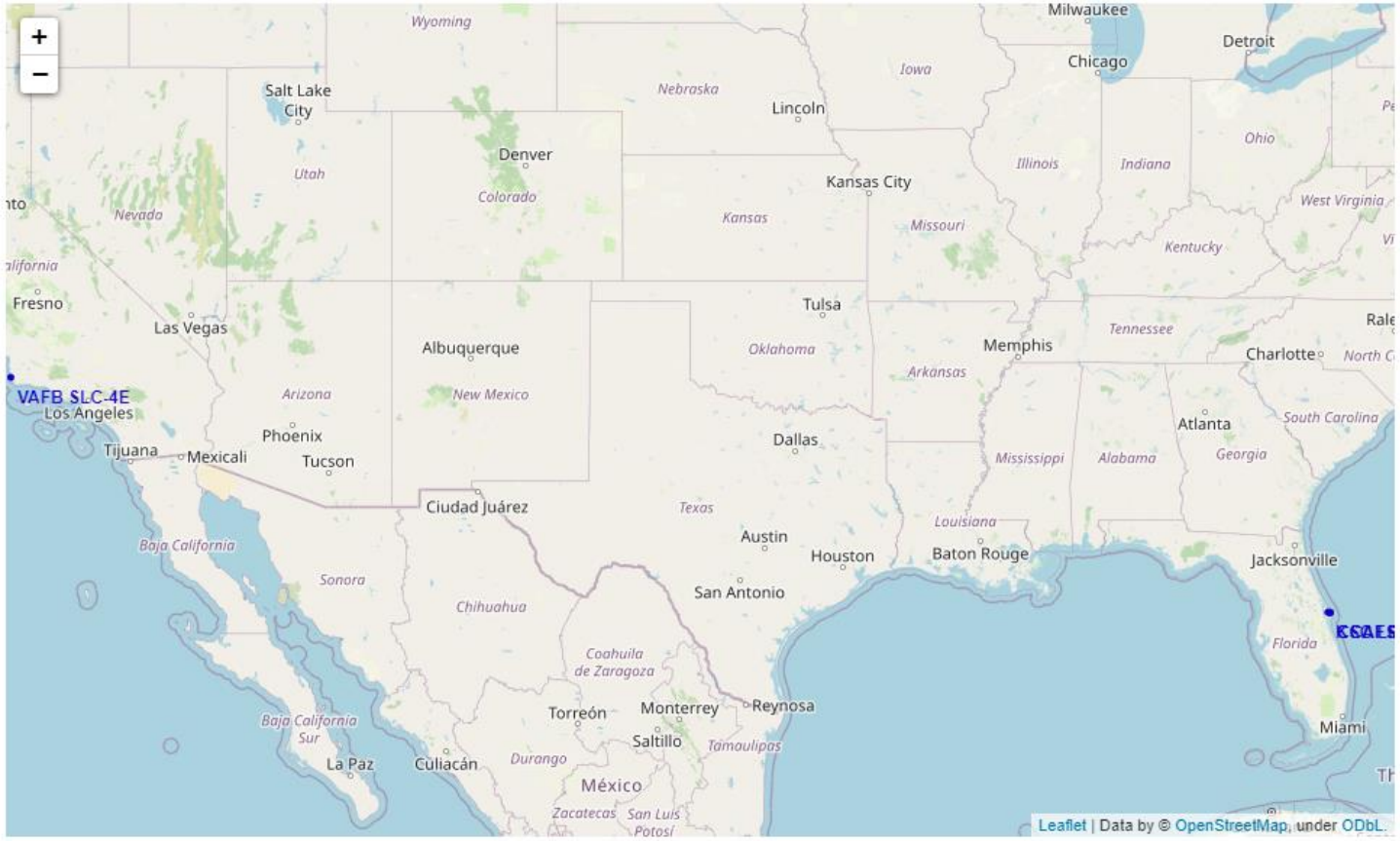
| landing__outcome | outcome_count |
|---|---|
| No attempt | 10 |
| Failure (drone ship) | 5 |
| Success (drone ship) | 5 |
| Controlled (ocean) | 3 |
| Success (ground pad) | 3 |
| Failure (parachute) | 2 |
| Uncontrolled (ocean) | 2 |
| Precluded (drone ship) | 1 |

- Count of success outcomes: 5 (drone ship), 3 (ground pad)

# Interactive map with Folium

# Launch Sites Marked on the World Map

Out[76]:




**Most of Launch sites considered in this project are in proximity to the Equator line.**

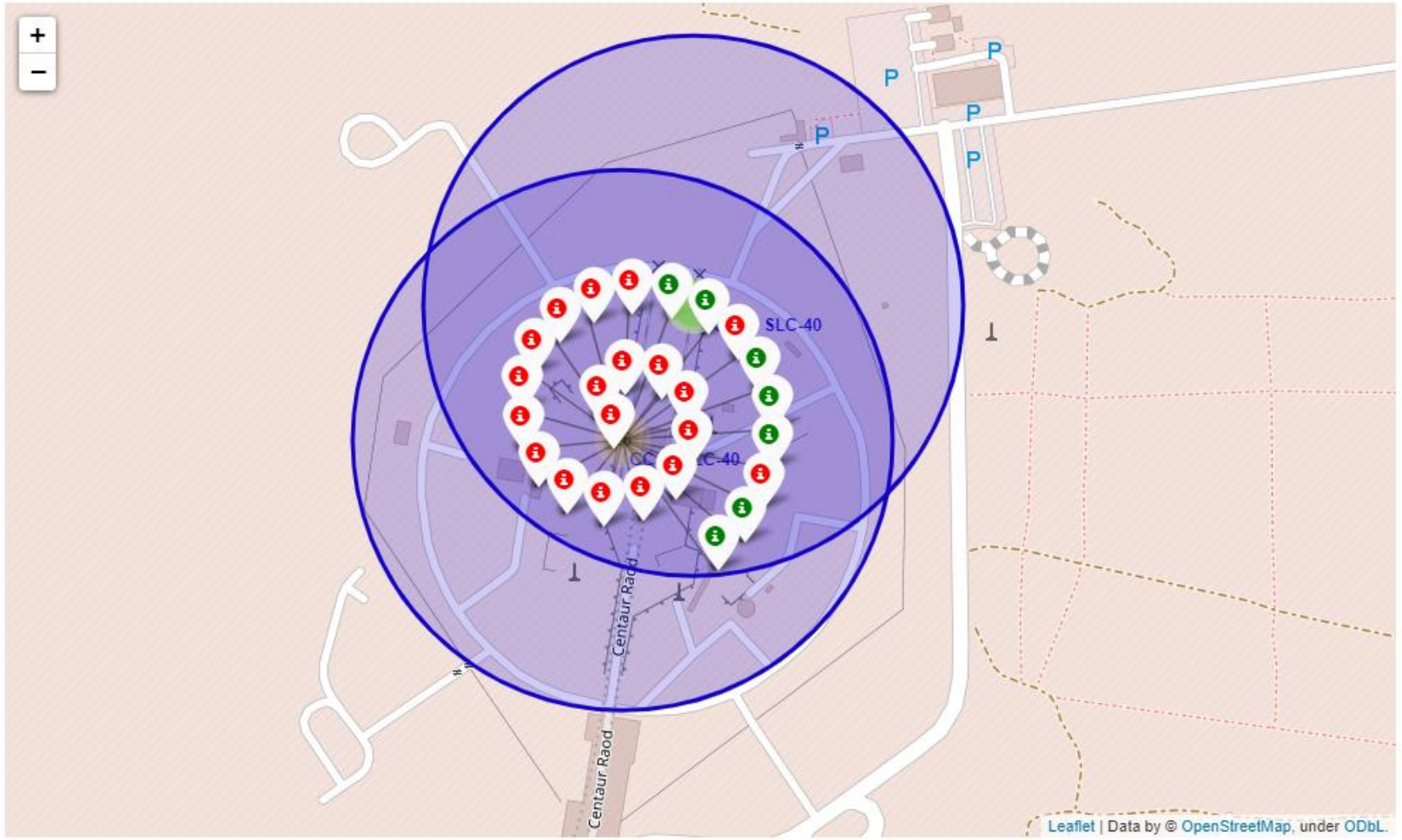
Launch sites are made at the closest point possible to Equator line, because anything on the surface of the Earth at the equator is already moving at the maximum speed (1670 kilometers per hour). For example launching from the equator makes the spacecraft move almost 500 km/hour faster once it is launched compared half way to north pole.

**All launch sites considered in this project are in very close proximity to the coast**

Starting rockets towards the ocean helps to minimise the risk of having any debris dropping or exploding near people.

# Success Rate for Launch Site (CCAFS LC-40)

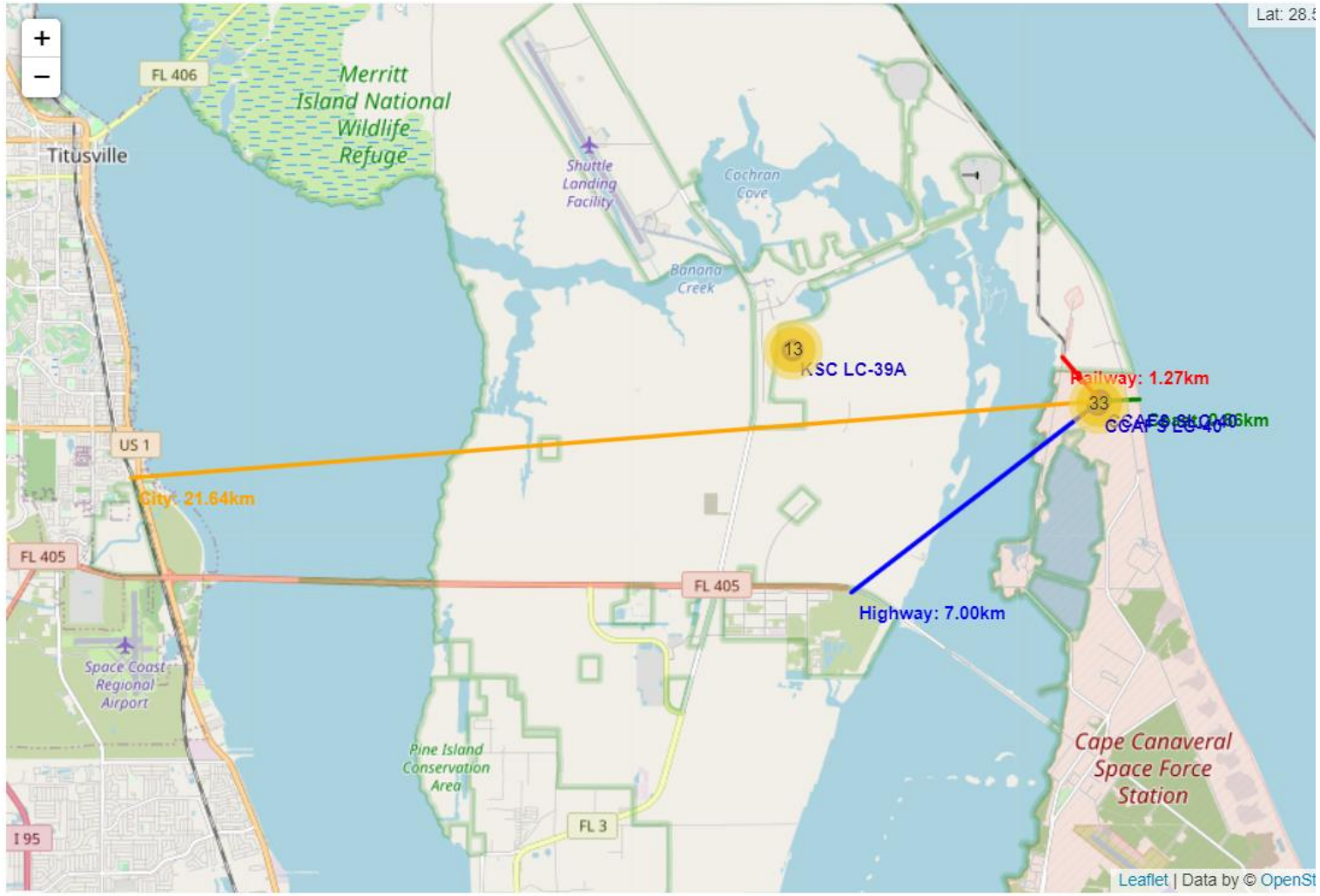
Out[82]:




**For the Launch Site CCAFS LC-40 the success rate is not very high**

# Distance from Launch Site (CCAFS LC-40) to Its Proximities

Out[88]:

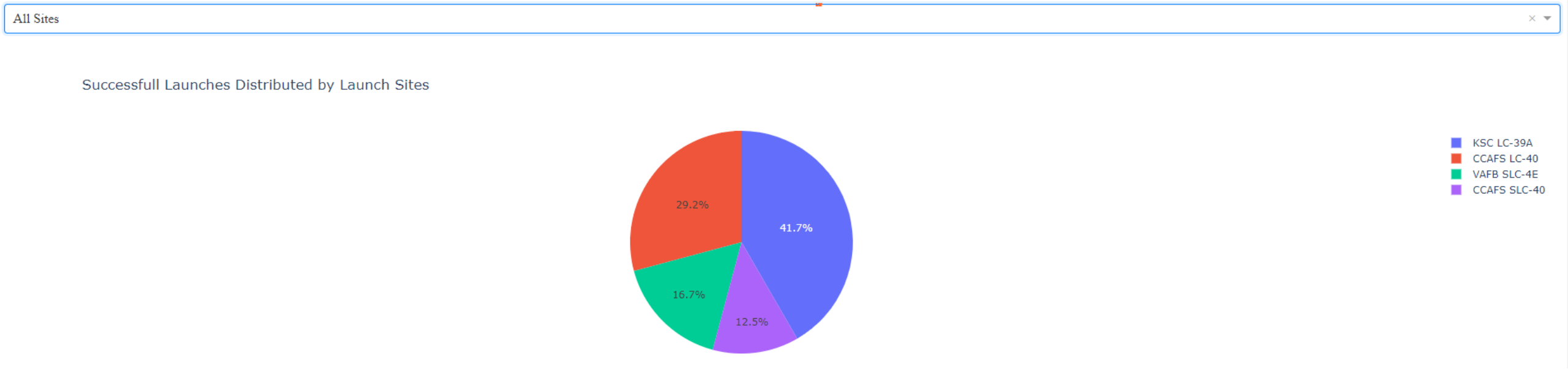



Distance to railway: 1,27km
Distance to highway: 7,00km
Distance to coastline: 0,86km
Distance to next city: 21,64km

- Launch site is built close to major bodies of water to ensure that no components are shed over populated areas.
- Launch site is built next to railways/highways to provide convinient transpotation of space-craft parts, cargos and stuff.
- A rocket launch site is built as far as possible away from major population centers in order to mitigate risk to bystanders should a rocket experience a catastrophic failure.

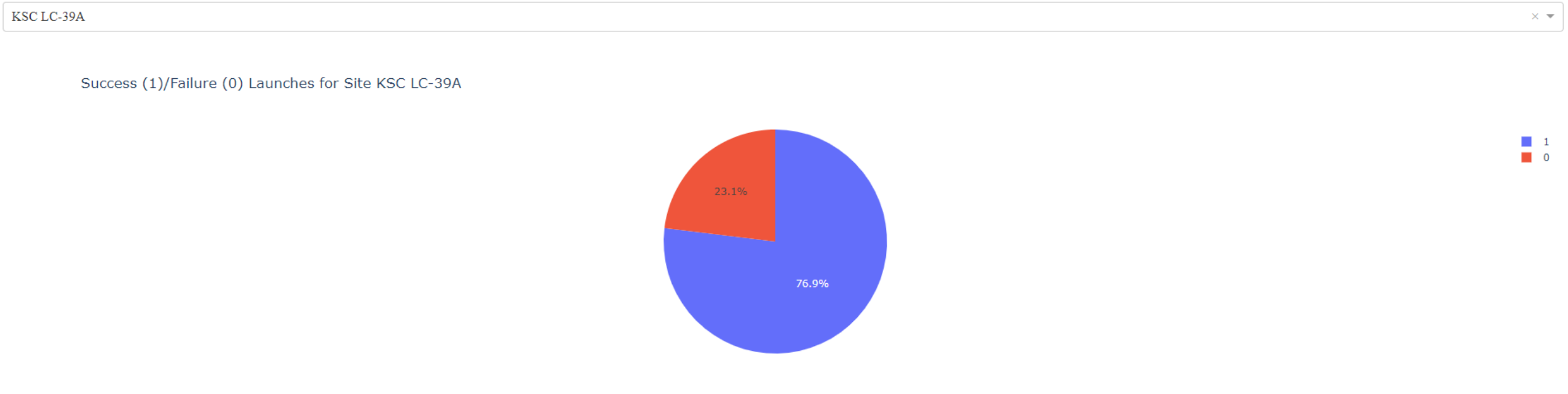
# Build a Dashboard with Plotly Dash

# Launch Success Count for All Sites




- Most of successful launches were made on KSC LC-39A launch site

# Success/Failure Rate For KSC LC-39A Launch Site

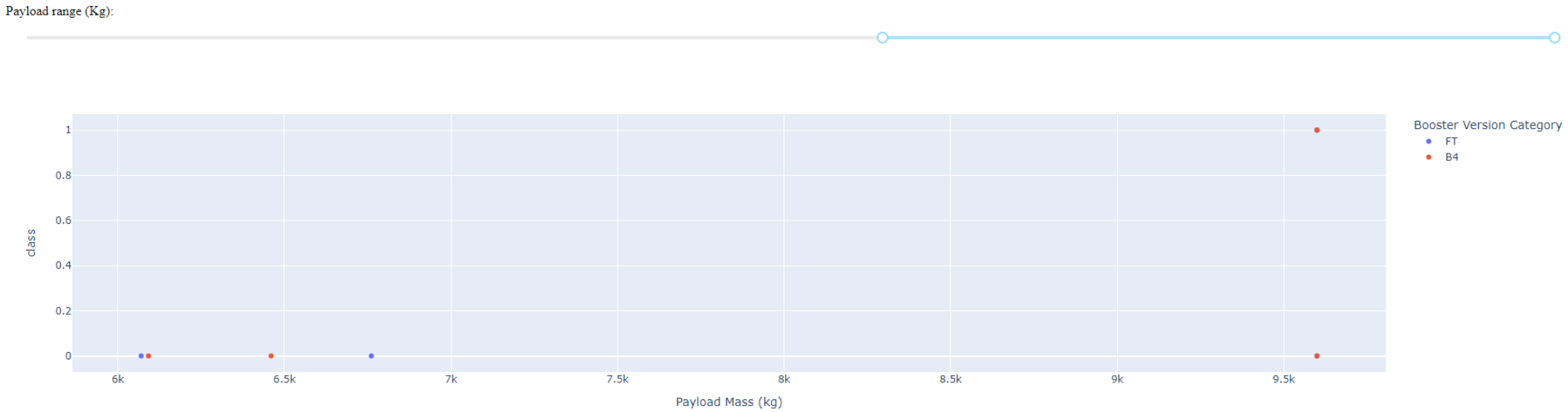



- KSC LC-39A launch site has the highest success rate (76,9%)

# Payload vs. Launch Outcome (Different Payload Ranges), All Launch Sites

Payload under 3 000kg:


Payload over 6 000kg:

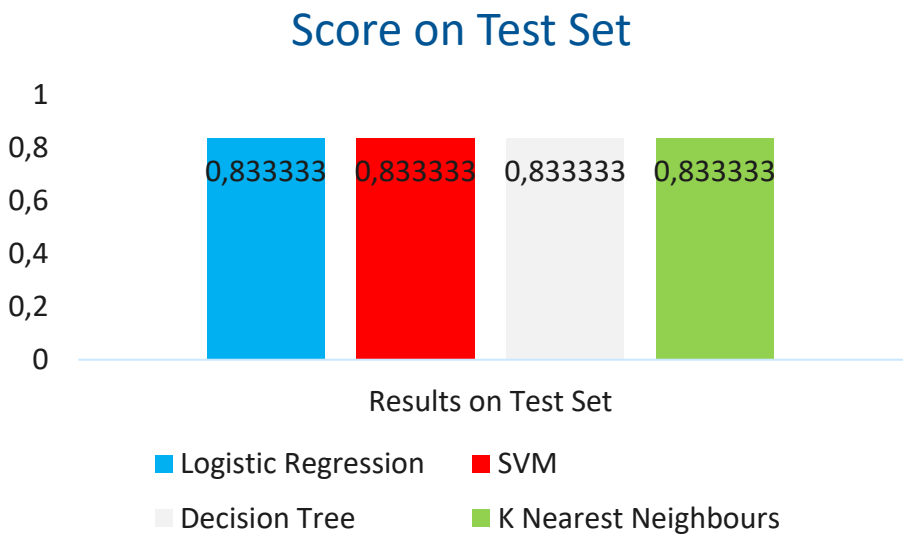


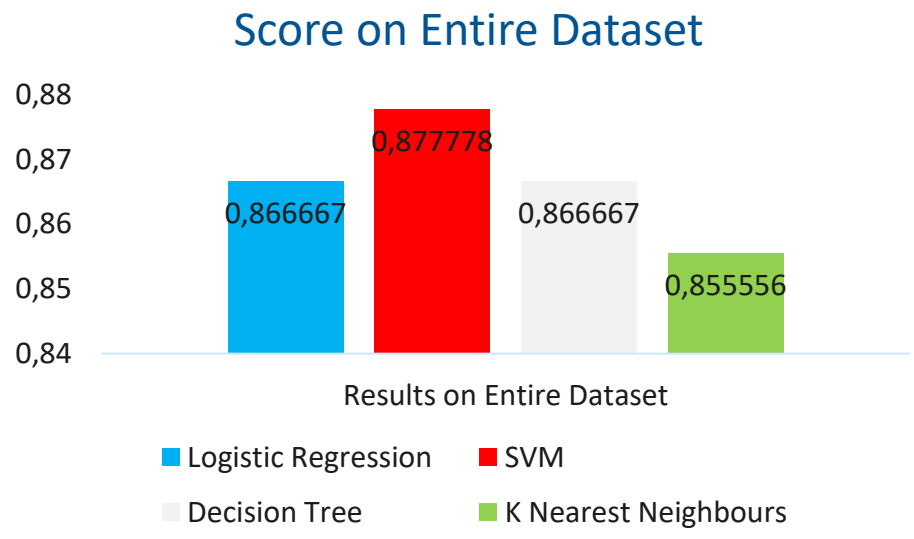
# Predictive analysis (Classification)

# Classification Accuracy

**Score comparison for different methods on Test Set and Entire Dataset**

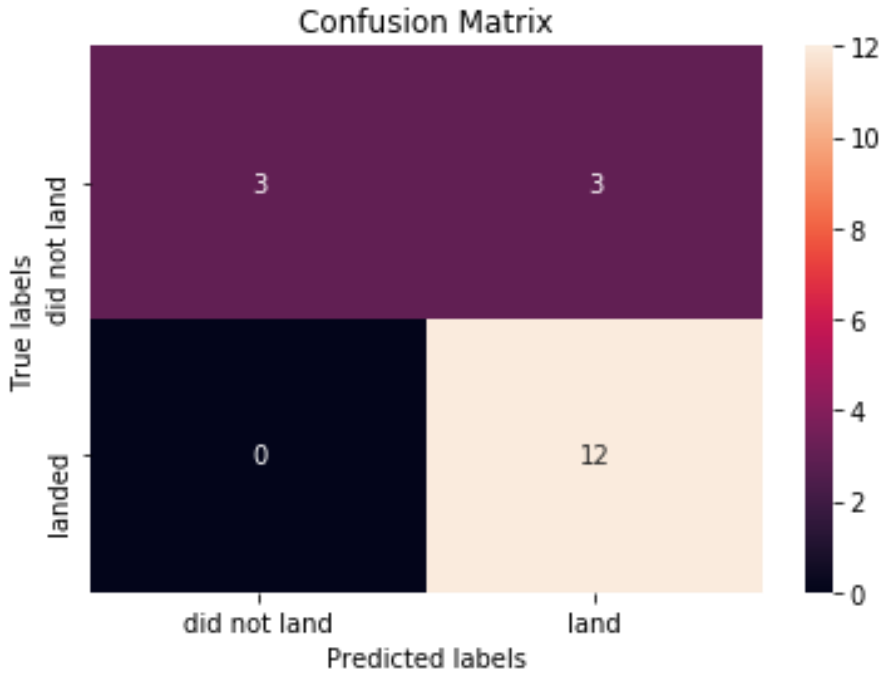
| Method | Test Dataset Score | Best Train Score | Whole Dataset (Train+Test) Score |
|---|---|---|---|
| **Logistic Regression** | 0.833333 | 0.846429 | 0.866667 |
| **SVM** | 0.833333 | 0.848214 | 0.877778 |
| **Decision Tree** | 0.833333 | 0.873214 | 0.866667 |
| **KNN** | 0.833333 | 0.848214 | 0.855556 |




- All prediction methods showed pretty high accuracy score (over 80%)
- All prediction methods showed equal accuracy score 83,33% on test set.
- SVM method performed best when making prediction on the entire dataset.

# Confusion Matrix for the Best Performing Method (SVM)




Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the major problem is false positives.

- SVM method can distinguish between the different classes
- False positives is the point for improvement of the prediction accuracy

# CONCLUSION

- Gathered datasets provide a good basis both for prediction if the Falcon 9 Space Rocket first stage will land successfully and for EDA concerning Space X rocket launches.
- Various EDA techniques show that a list of different factors affect the success of first stage landing.
- High accuracy score (over 80%) could be achieved for predictions based on gathered datasets.
- SVM method provides best accuracy score result (on entire dataset).
- False positives rate is the point for further improvement of the predicting model based on confusion matrix analysis.

# APPENDIX

## Fixed code line to fill the “Customer” column from Wikipedia (Week 1, Web scraping lab):

```
# Customer
# TODO: Append the customer into launch_dict with key `Customer`
try:
    customer = row[6].a.string
except:
    customer = 'Various'
launch_dict['Customer'].append(customer)
#print(customer)
```