IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Raimundo Nonato da Cruz
19/04/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Collecting the Data from an API (SpaceX Rest API).
  - Collecting the Data with Web Scraping.
  - Data Wrangling.
  - Exploratory Data Analysis using SQL (SQLite).
  - Exploratory Data Analysis for Data Visualization.
  - Interactive Visual Analytics and Dashboard (Folium and Plotly Dash).
  - Predictive Analysis (Machine Learning).

Executive Summary

- Summary of all results
  - Requested SpaceX API (rockets, launchpads, payloads and cores) and handling datasets.
  - Requested the Falcon9 Launch Wiki page, Extract all column/variable names from HTML and create a data frame by parsing the launch HTML.
  - Calculated the number of launches on each site, calculated the number and occurrence of each orbit, calculated de number and occurrence of mission outcome per orbit type and created a landing outcome label from Outcome column.
  - Connected to SQLite and was executed SQL queries.
  - Exploratory Data Analysis and visualized the relationship between Flight Number and Launch Site, relationship between Payload and Launch Site, relationship between success rate of each orbit type, relationship between Flight Number and Orbit type, relationship between Payload and Orbit type and the launch success yearly trend
  - Generated map with Python and marked all launch site on map, marked the success/failed launches for each site on the map and Calculate the distances between a launch site to its proximities (with Folium). Success Count for all launch sites (SpaceX Launch Record Dashboard.
  - Machine Learning Prediction used to create a column for the class, Standardize the data and. Split into training data and test data. Find best Hyperparameter for SVM, Classification Trees and Logistic Regression. Find the method performs best using test data

Introduction

- Project background and context

The commercial space age is here, companies are making space travel affordable for everyone. Virgin Galactic is providing suborbital spaceflights. Rocket Lab is small satellite provider. Blue Origin manufactures sub-orbital and orbital reusable rockets. Perhaps the most successful is SpaceX. SpaceX's accomplishments include: Sending spacecraft to the international Space Station. Starlink a satellite internet constellation providing satellite internet access. Sending manned mission to space.

One reason SpaceX can do this is the rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefor, if we can determine if the first stage will land, we can determine the cost of a launch.

The objective of this work is to determine through artificial intelligence if the first stage of the rocket can be reused.

Introduction

- Problems you want to find answer
  - Can recover the first stage?
  - Sometimes the first stage does not land?
  - Sometimes it will crash?
  - Can SpaceX sacrifice the first stage due to the mission parameters like payload, orbit, and customer?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  Data was collected through request to SpaceX API :

  "https://api.spacexdata.com/v4/rockets/"+str(x)).json()

  "https://api.spacexdata.com/v4/launchpads/"+str(x)).json()

  "https://api.spacexdata.com/v4/payloads/"+load).json()

  "https://api.spacexdata.com/v4/cores/"+core['core']).json()

  Data was collected through web scraping:

  https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Methodology

Executive Summary

- Perform data wrangling

I perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully lands to a ground. False RTLS means the mission outcome was unsuccessfully lands to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship. False ASDS means the mission was unsuccessfully landed on a drone ship.

Those outcomes was converted mainly into Training Labels with 1 (one) means the booster successfully landed and 0 (zero) means it was unsuccessfully.

Methodology

Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL

  First, I downloaded the dataset, connected to database SQLite and executed SQL queries to solve assignments, like as unique Launch Site. Launch Site beginning with 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.

Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash
- Using Folium to interactive visual analytics.
  - Generating maps with Python to mark all launch sites on a map; mark the success/failed launches for each site on the map and calculate the distances between a launch site to its proximities.
- Using Plotly Dash to interactive visual analytics.
  - Pandas dash.
  - wget"https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_dash.csv"
  - wget"https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_3/spacex_dash_app.py"
  - In spacex_dash_app.py was added Launch Site Drop-down Input Component; added a callback function to render based on select site dropdown; added a range slider to select payload and added a callback function to render a scatter plot.

Methodology

Executive Summary

- Perform predictive analysis using classification models
    - Machine Learning
    - Preprocessing
    - Train/test split
    - Model
    - Logistic Regression
    - Support Vector Machine
    - Decision Tree Classifier
    - K – nearest neighbors
    - Confusion Matrix

Data Collection

- Describe how data sets were collected.
    - requests.get(https://api.spacexdata.com/v4/rockets/"+str(x)).json())
    - requests.get(https://api.spacexdata.com/v4/launchpads/"+str(x)).json())
    - requests.get(https://api.spacexdata.com/v4/payloads/"+load).json())
    - requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json())
    - HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response

Data Collection – SpaceX API

- **SpaceX REST API.**
- **This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.**

Request and parse the SpaceX launch data using the GET request → Filter the dataframe to Only include Falcon 9 launches → Dealing with Missing Values

https://github.com/Nonato57/Space-YYY/blob/master/Data_Collection_API.ipynb

Data Collection - Scraping

- Web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled “List a Falcon 9 and Falcon Heavy Launches”

HTTP GET method to request the Falcon9 Launch HTML page

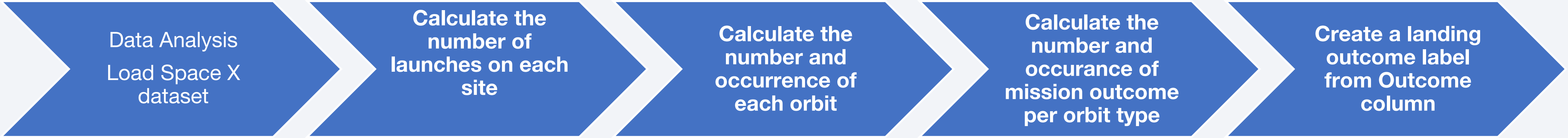
Extract all column/variable names from the HTML table

Create a data frame by parsing the launch HTML tables

https://github.com/Nonato57/Space-YYY/blob/master/jupyter-labs-webscraping.ipynb

Data Wrangling

- Perform exploratory Data Analysis and determine Training Labels
    - Exploratory Data Analysis
    - Determine Training Labels



https://github.com/Nonato57/Space-YYY/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Chart Plotted:

- Flight Number vs Payload Mass: How the FlightNumber and Payload variables would affect the launch outcome.
- Flight Number vs Launch Site: How the FlightNumber and Launch Site variables would affect the launch outcome.
- Payload vs Launch Site: How the Payload Mass affect the launch outcome.
- Success Rate vs Orbit Type: How the Orbit Type affect the Success Rate.
- Payload vs Orbit Type: Analyze if Payload Mass affect the choice the Orbit Type.
- Success Yearly Trend: Success Rate per Year

https://github.com/Nonato57/Space-YYY/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

SQL queries performed

- %sql select distinct Launch_Site from SPACEXTBL
- %sql select * from SPACEXTBL where Launch_Site like ‘CCA%’ limit 5
- %sql select sum(payload_mass__kg_) from SPACEXTBL where customer = ‘NASA (CRS)’
- %sql select avg(payload_mass__kg_) from SPACEXTBL where booster_version = ‘F9 v1.1’
- %sql select min(DATE) from SPACEXTBL where “Landind _Outcome”) = ‘Success (ground pad)’
- %sql select booster_version from SPACETBL where Landind _Outcome” = ´Success (drone ship)’ and payload_mass__kg_ between 4000 and 6000
- %sql select mission_outcome, count(mission_outcome) from SPACEXTBL group by mission_outcome

EDA with SQL

SQL queries performed

- %sql select booster_version, payload_mass__kg_ from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
- %sql select booster_version, lauch_site from SPACEXTBL where "Landing _Outcome" = 'Failure drone ship)' and substr(Date, 7,4) = '2015'
- %sql select count("Landing _Outcome"), "Landing _Outcome" from SPACETBL where DATE between '04-06-2010' and '20-03-2017' group by "Landing _Outcome' order by count("Landing _Outcome") desc

https://github.com/Nonato57/Space-YYY/blob/master/jupyter-labs-eda-sql-coursera_sqlite.ipynb

Build an Interactive Map with Folium

- Circle to add a highlighted circle area with a text label on a specific coordinate.
- Marker to put a text label
- Marker Cluster
- Mouse Position
- Marker Color
- Polyline

The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations

https://github.com/Nonato57/Space-YYY/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Plots/graphs and interactions added to a dashboard
  - Dropdown list to enable Launch Site selection.
  - Pie chart to show the total successful launches count for all sites.
  - Slider to select payload range.
  - Scatter chart to show the correlation between payload and launch success.
  - Callback function for `site-dropdown` as input, `success-pie-chart` as output.
  - Callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output.

https://github.com/Nonato57/Space-YYY/blob/master/Spacex_Dash_Screeshot.PNG

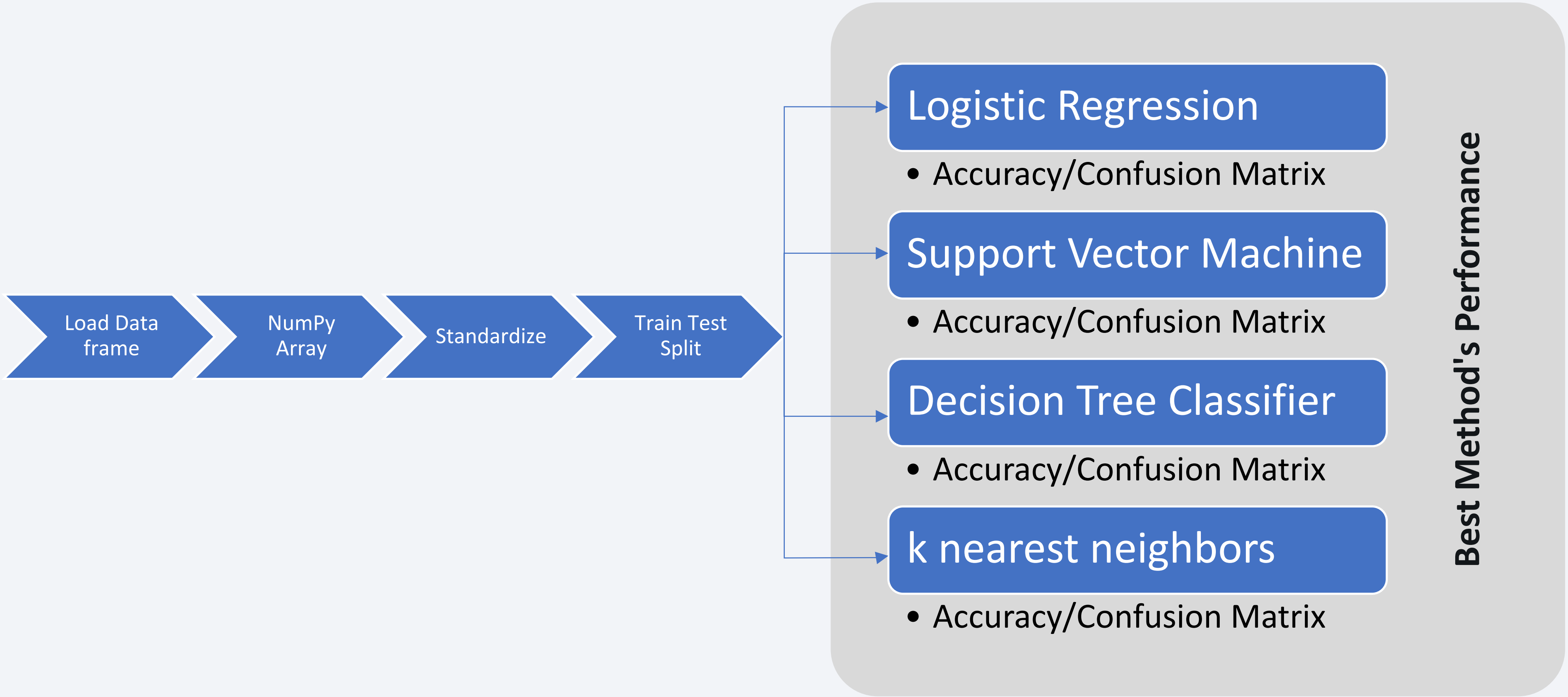
Predictive Analysis (Classification)

- Load the data frame, accessing csv file.
- Create a NumPy array from the column Class in data.
- Standardize the data in X then reassign it to the variable X.
- Function train_test_split to split the data X and Y into training and test data.
- Create a logistic regression object then create a GridSearchCV object logreg_cv with cv = 10.
- Calculate the accuracy on the test data using the method score. Plot Confusion Matrix
- Create a support vector machine object then create a GridSearchCV object svm_cv with cv – 10.
- Calculate the accuracy on the test data using the method score. Plot Confusion Matrix.
- Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10.

Predictive Analysis (Classification)

- Calculate the accuracy of tree_cv on the test data using the method score. Plot Confusion Matrix.
- Create a k nearest neighbors object then create a GridSearchCV object knn_cv with cv = 10.
- Find the method performs best (Accuracy).

Predictive Analysis (Classification)



https://github.com/Nonato57/Space-YYY/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

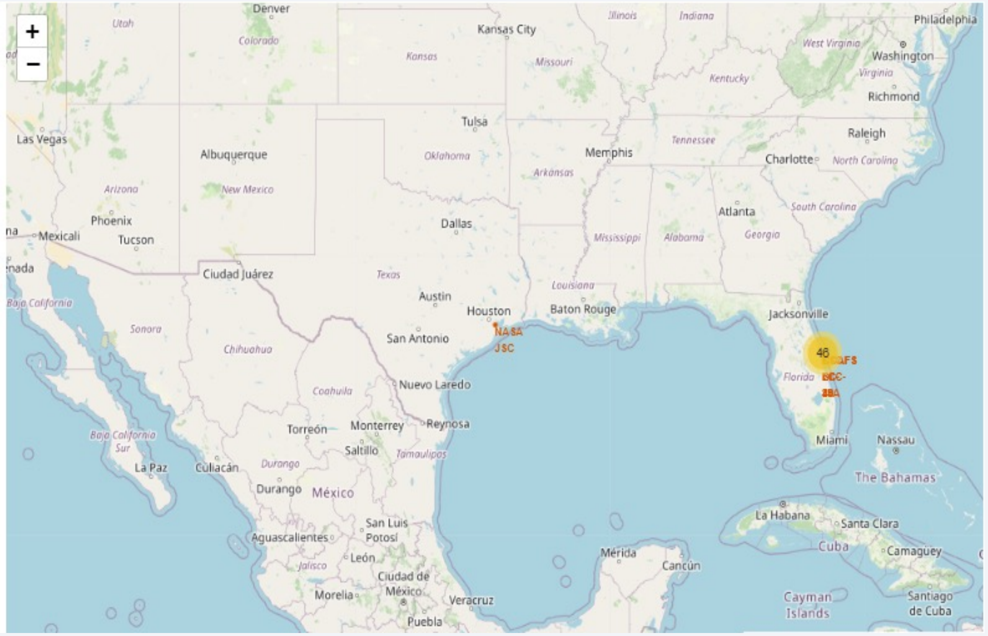
Results

- Exploratory data analysis results
  - SpaceX used 4 launch sites: CCAFS LC-40, VAFB SLC-4E, KSC LC-39A AND CCAFS SLC-40,
  - Space X made the first launch with NASA: 04-06-2010.
  - Total payload mass carried by NASA: 45.596kg
  - Average Payload mass carried by F9 v1.1: 2928.4 kg
  - Date of first successful landing outcome in ground pad: 01-05-2017
  - Boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000: F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2.
  - 12 Booster Version carried maximum payload mass (15.000 kg).
  - Launching Outcome failure with done ship: F9 v1.1B1012 and F9 v1.1 B1015, both at the launch site CCAFS LC-40.
  - Success landing outcome between 04-06-2010 and 20-03-2017: 34 (Success, Success drone ship and Success ground pad.

Results

- Interactive analytics demo in screenshots

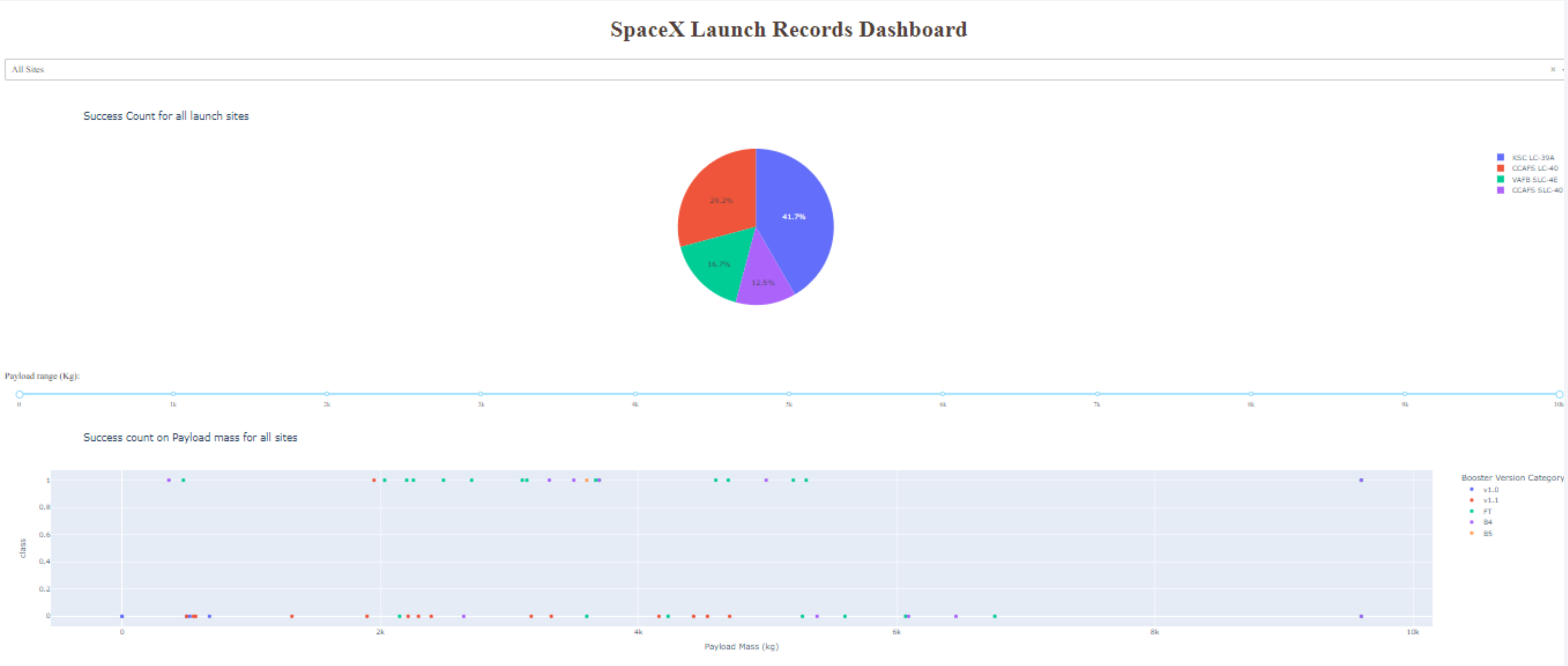
With Folium: Mark the success/failed launches for each site on the map



Results

- Interactive analytics demo in screenshots

With Ploty Dash:



Results

- Predictive analysis results

  - Logistic regression: tuned hyperparameters :(best parameters) {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
  - accuracy : 0.8464285714285713. Score: 0.8333333333333334

  - Support Vector Machine: tuned hyperparameters :(best parameters) {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'} accuracy : 0.8482142857142856. Score : 0.8333333333333334.

  - Decision tree classifier: tuned hyperparameters :(best parameters) {'criterion': 'gini', 'max_depth': 14, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'}
  - accuracy : 0.8892857142857145. Score: 0.6666666666666666.

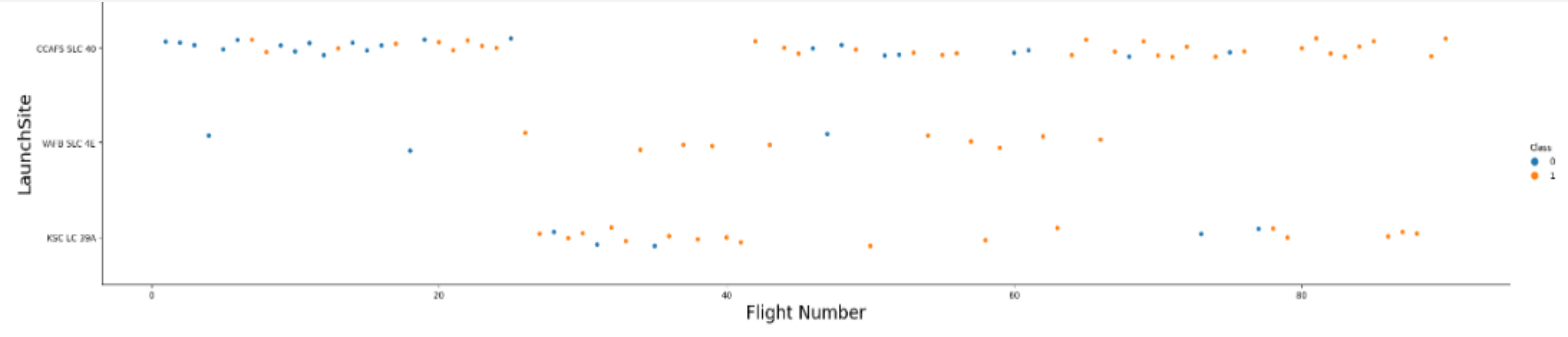
  - k nearest neighbors: tuned hyperparameters :(best parameters) {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
  - accuracy : 0.8482142857142858. Score: 0.8333333333333334

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Scatter plot of Flight Number vs. Launch Site

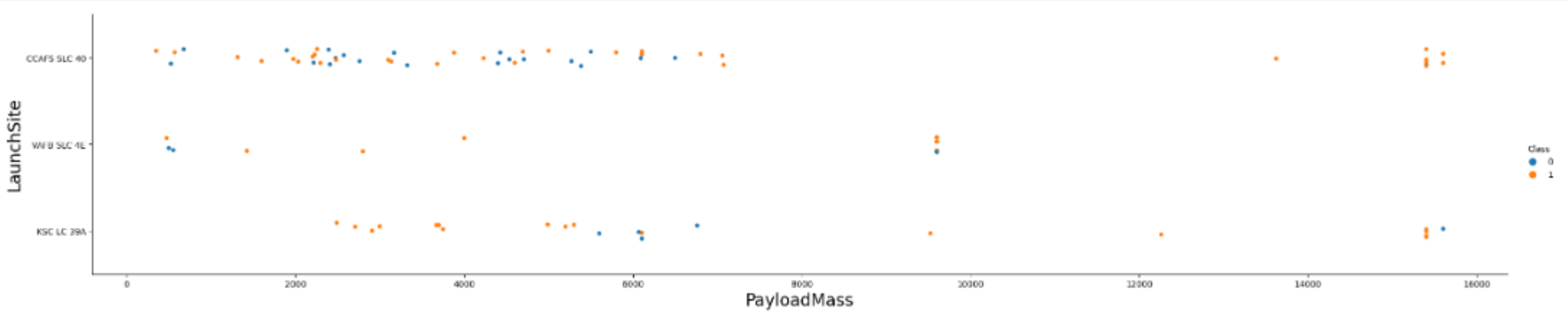


- Screenshot of the scatter plot with explanations

```
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.
Most of the flight numbers took place at launch site CCAFS SLC-40. At first there were failures, but the last few launches were
successful.
```

Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site

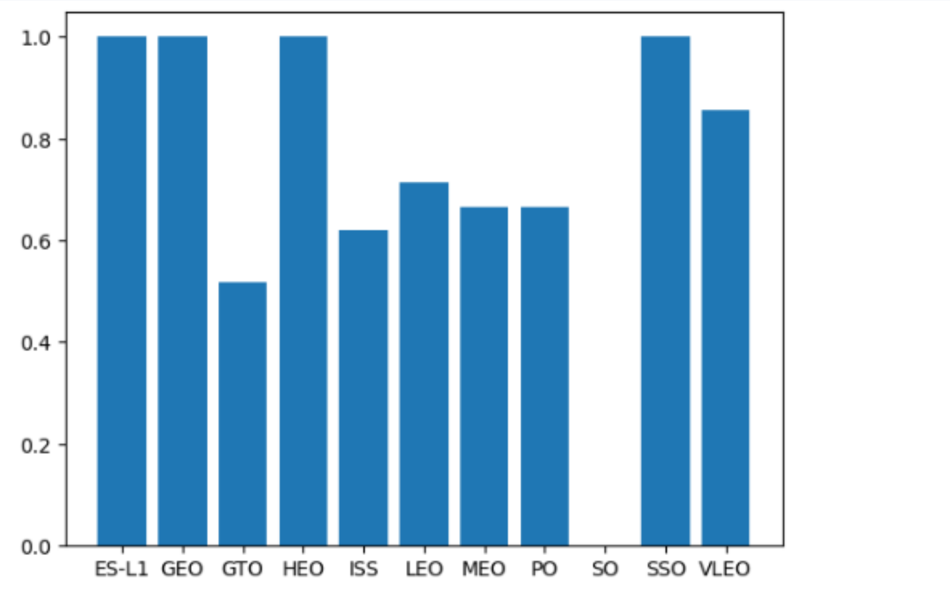


- Screenshot of the scatter plot with explanations

Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

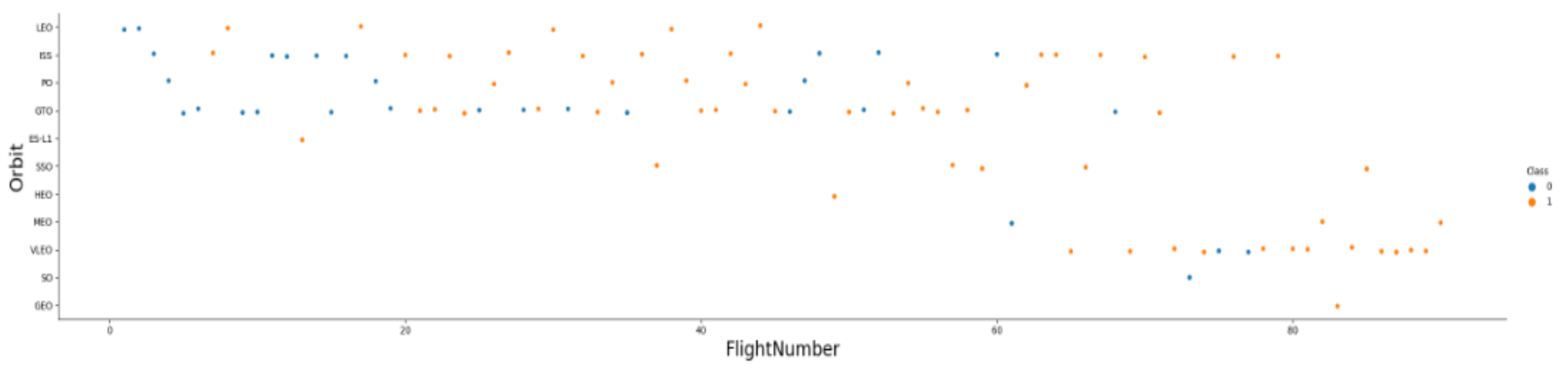


- Screenshot of the bar chart with explanations

```
Analyze the ploted bar chart try to find which orbits have high sucess rate.
The orbits ES-L1, GEO, HEO and SSO have high sucess rate.
```

Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

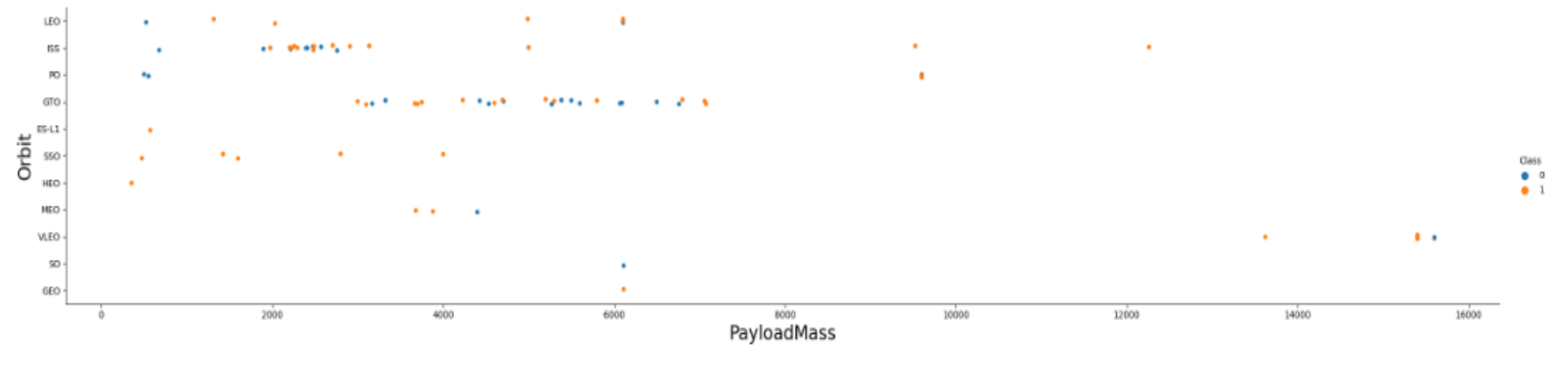


- Screenshot of the scatter plot with explanations

You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type



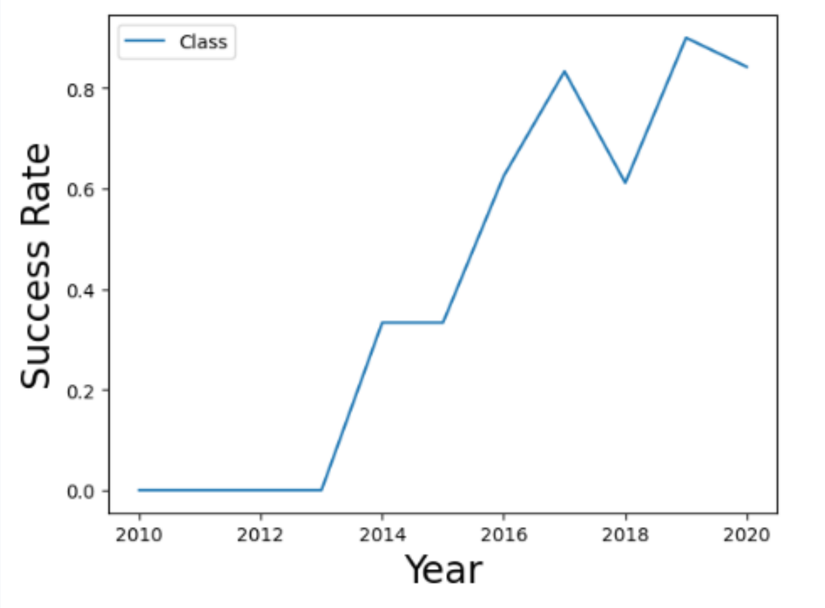
- Screenshot of the scatter plot with explanations

With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

- Show a line chart of yearly average success rate



- Screenshot of the lie chart with explanations

you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- Find the names of the unique launch sites

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

```
%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

SpaceX used 4 launch site to lauching rockets.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

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

```
%sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5
```

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

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

Quey shows the first 5 records of CCAFS LC-40. But there are Other launch site as CCAFS SLC-40.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

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

Query shows the total payload carried (45.596kg) to NASA (CRS) customer.

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

```
%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

Query shows the average payload mass (2928.4 kg) carried by booster version F9 v1.1

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

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

Hint:Use min function

```
%sql select min(DATE) from SPACEXTBL WHERE "Landing _Outcome" = 'Success (ground pad)'
#%sql select "Landing _Outcome" from SPACEXTBL
```

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

min(DATE)
01-05-2017

The date of the first sucessful landind outcome is 01-05-2017.

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had 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

```
%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.
```

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

Were found 4 names of boosters which had successfully landed.

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

```
%sql select mission_outcome, count(mission_outcome) from SPACEXTBL GROUP BY mission_outcome
```

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

Mission_Outcome	count(mission_outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Were found 100 succesful and 1 failure mission outcomes.

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

```
%sql select booster_version, payload_mass__kg_ from SPACEXTBL\
where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

Were found 12 names of booster which carried the maximum payload mass (15600kg)

2015 Launch Records

- List the failed landing outcomes in drone ship, their booster versions, and launch site names for 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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%sql select booster_version, launch_site from SPACEXTBL where "Landing _Outcome" = 'Failure (drone ship)' and substr(Date,7,4) = '2015'
```

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

Booster_Version	Launch_Site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

Were found 2 booster version, both CCAFS LC-40 launch site, that has failed in 2015.

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

- 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

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

```
%sql select count("Landing _Outcome"), "Landing _Outcome" from SPACEXTBL \
where DATE between '04-06-2010' and '20-03-2017' group by "Landing _Outcome"\
order by count("Landing _Outcome") desc
```

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

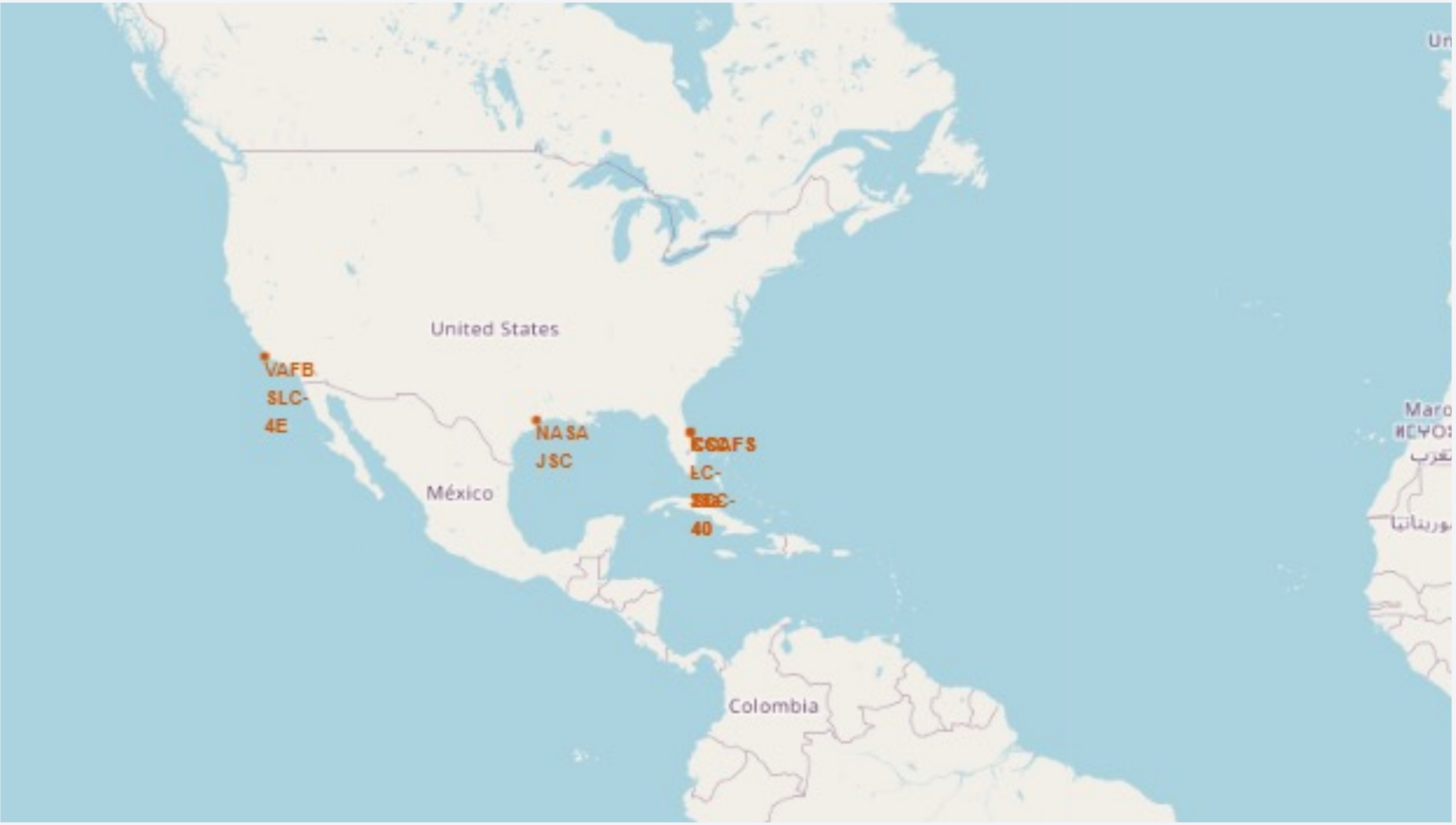
count("Landing _Outcome")	Landing _Outcome
20	Success
10	No attempt
8	Success (drone ship)
6	Success (ground pad)
4	Failure (drone ship)
3	Failure
3	Controlled (ocean)
2	Failure (parachute)
1	No attempt

Query ranked 9 landings outcome in descending order in the period of 04-06-2010 and 20-03-2017. Were found 34 successfull landings.

Section 3

Launch Sites Proximities Analysis

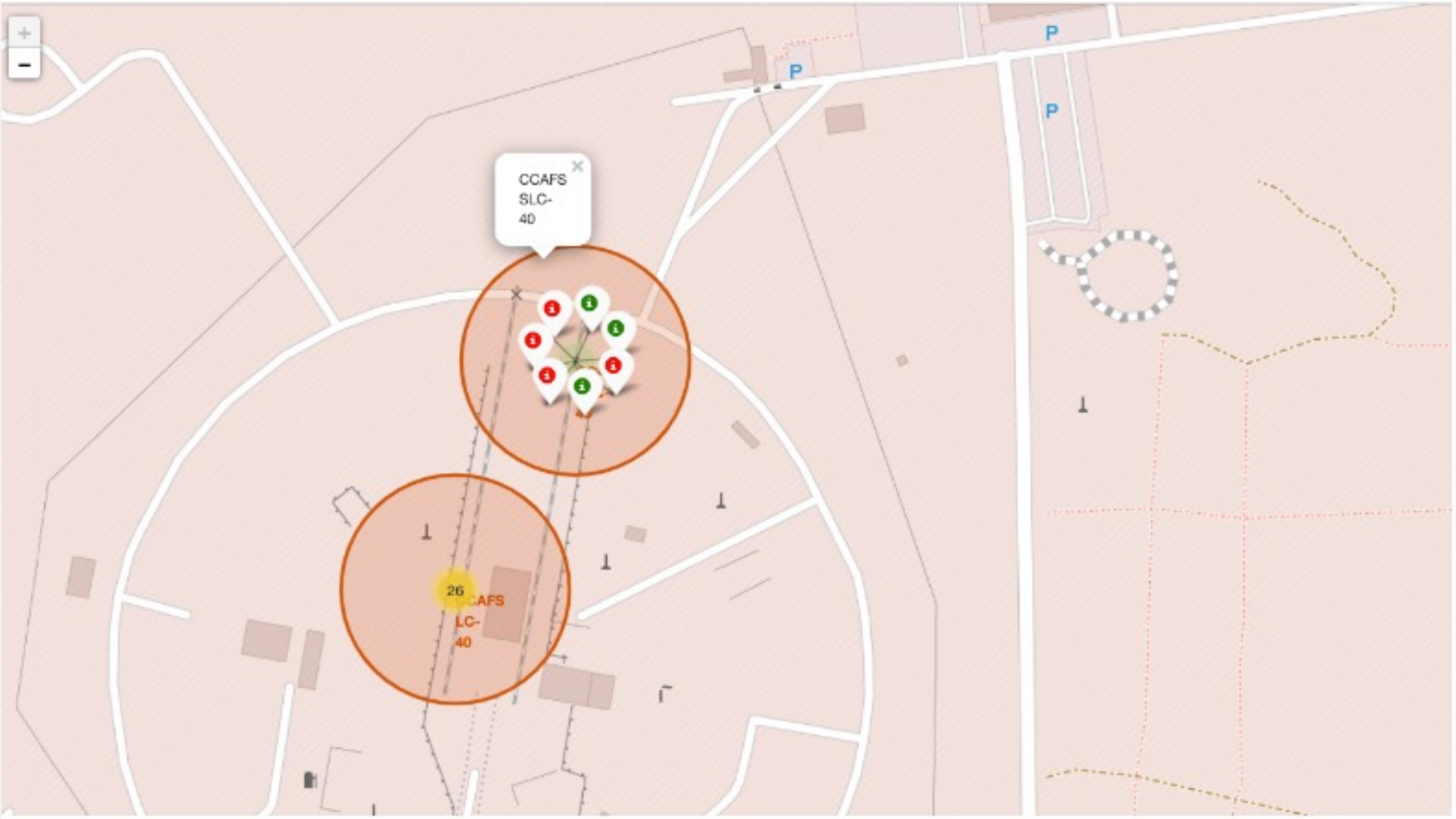
All launch sites on the map



Were used the folium circle and folium marker to mark all launch sites on the map.

Small yellow circle near the city and can zoom-in to see a larger circle

Mark on map success or unsuccess from color-labeled



From the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates.

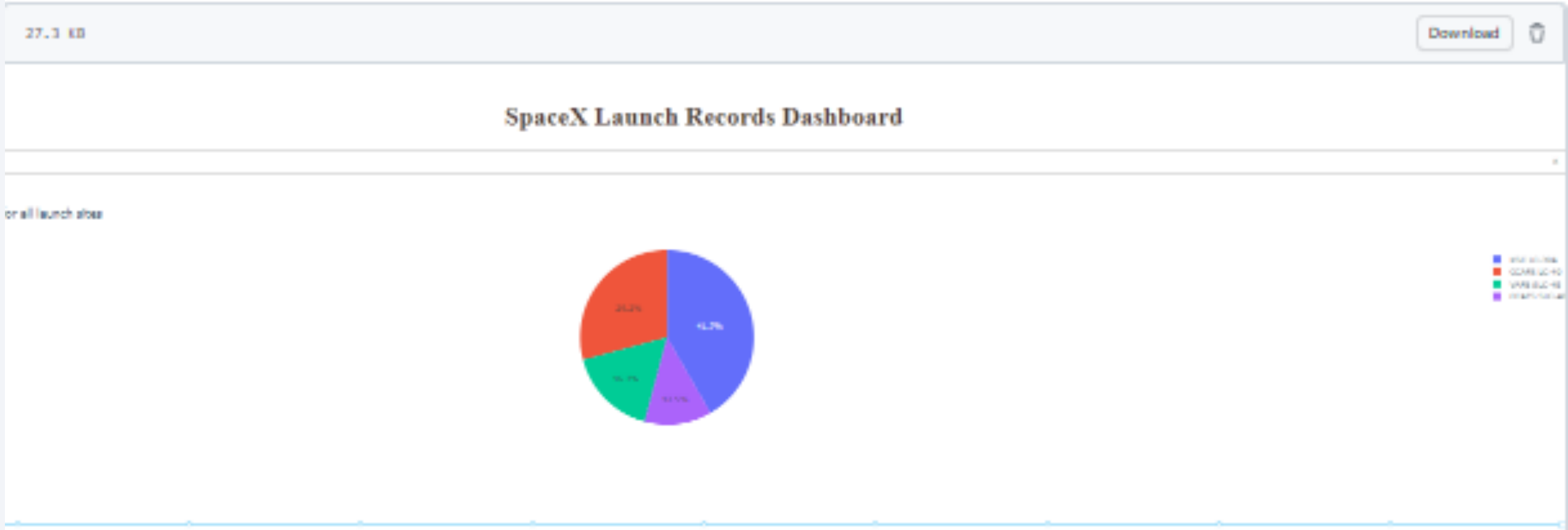
Calculate the distances between a launch site to its proximities

Using the Mouse Position It was found the distances between launch site (CCAFS SLC-40) and Coastline = 0.90km

Section 4

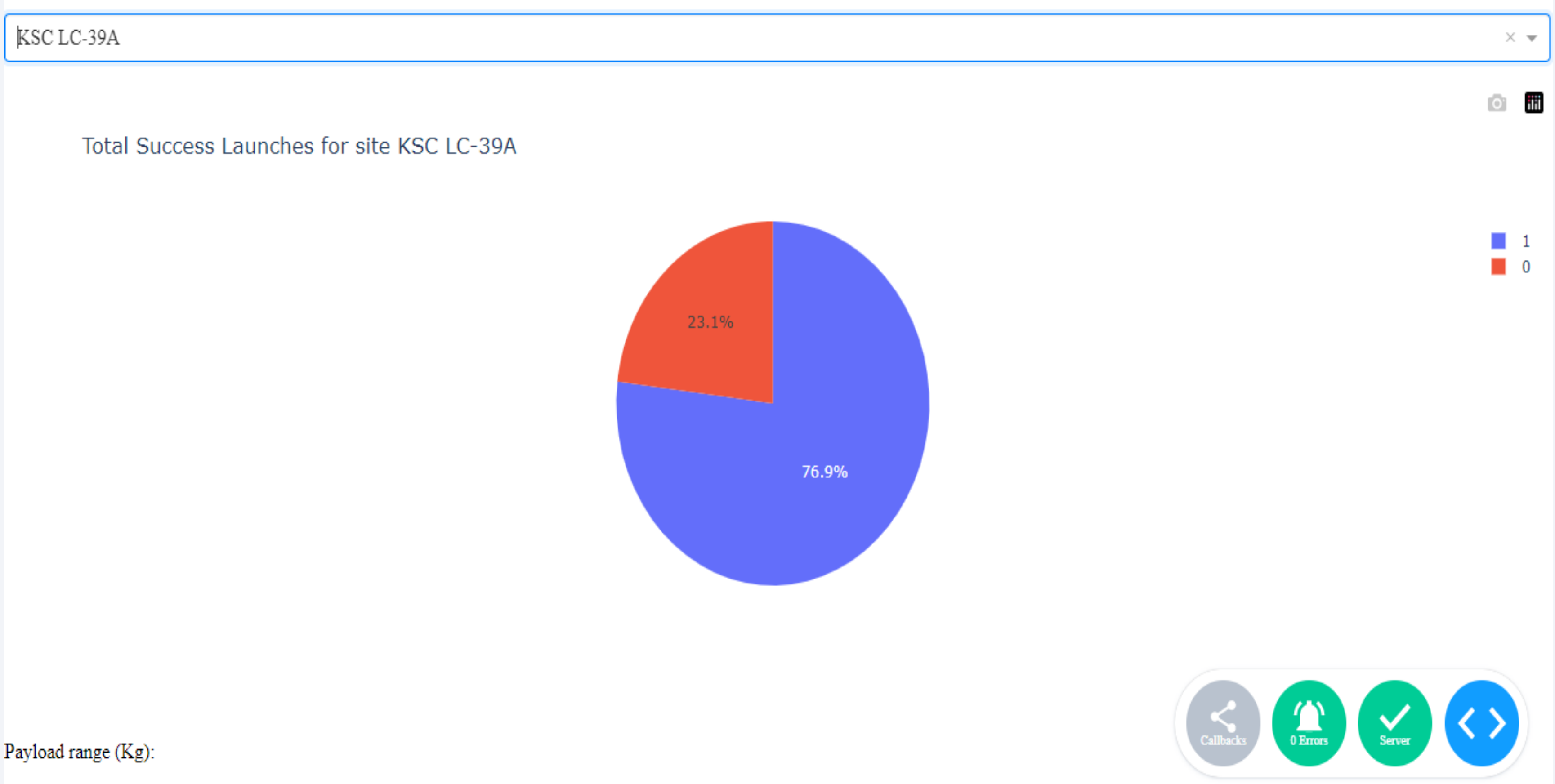
Build a Dashboard with Plotly Dash

SpaceX Launch Records Screenshot



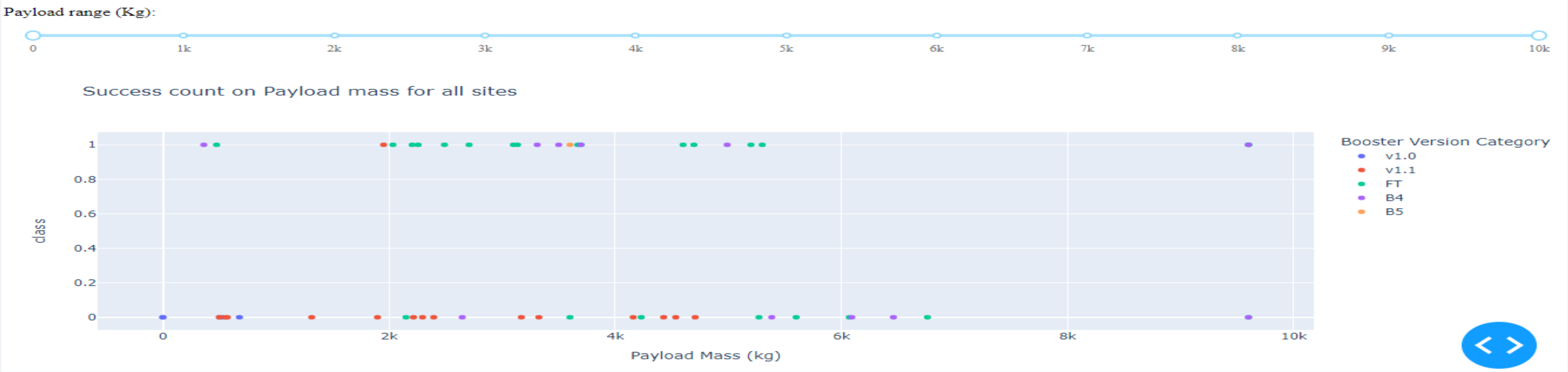
Was used a dropdown list to choose the launch site. A Pie chart to show the total successful launch count for all sites.
A Slider to select the payload range.
Callback function for `site-dropdown` as input, `success-pie-chart` as output .

Launch site with highest launch success ratio



We used the dropdown to choose the launch site with highest launch success ratio (KSC LC-39A)

Success count on payload mass for all sites



Success count on payload mass for all sites

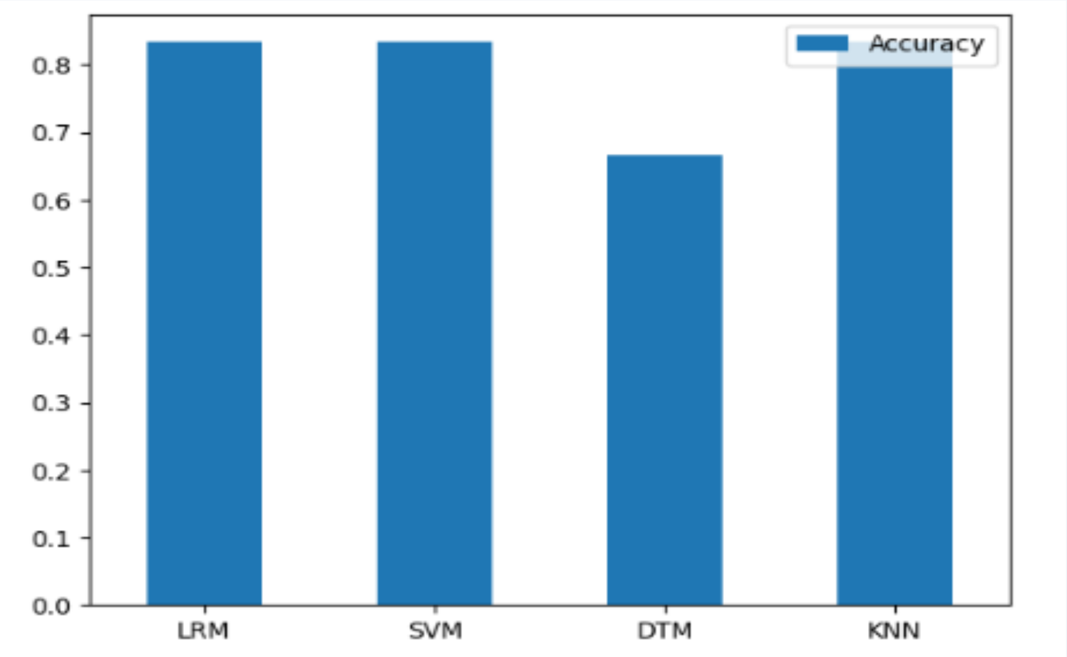
- In the first scatter plot, were found the payload mass for all sites.
- On the second scatter plot, we marked 2k in the payload range.

We used the range slider to select a payload mass
The use of a call-back function to render the scatter plot.
On payload range of 2k to 10k, the booster version categories are V1.1, FT, B4 and B5.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

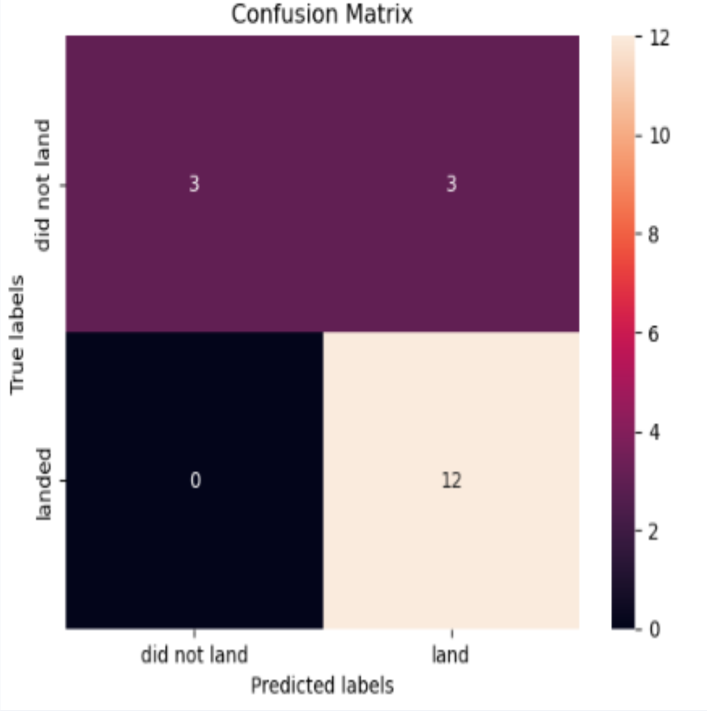


Were found 3 models that had the same accuracy.

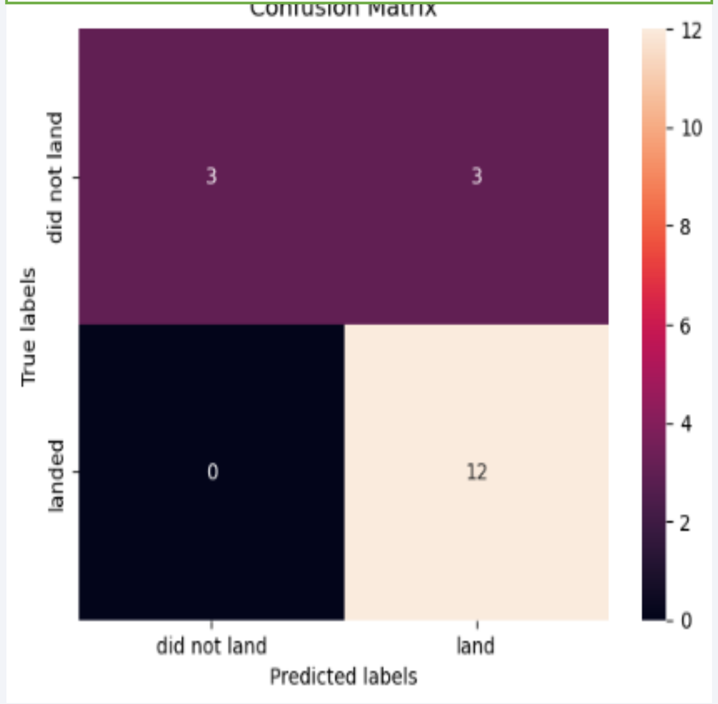
Confusion Matrix

- Confusion matrix of the best performing model with an explanation

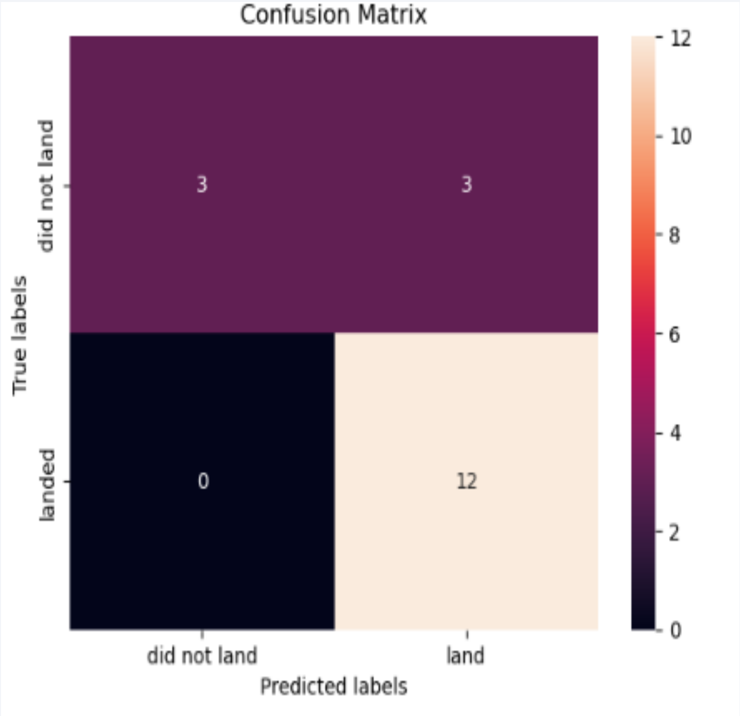
Logistic Regression



Support Vector Machine



KNN



Logistic Regression, SVM and KNN had the same accuracy, as we saw in confusion matrix.

Conclusions

- Importance of using IBM Watson for project development and sharing on GitHub.
- Usage of the REST API to get release data and turn it into JSON files. Convert the JSON file into a data frame and use it as a table to predict whether SpaceX will try to land a rocket or not.
- Usage of the Python BeautifulSoup package to web scraping the HTML tables that contain valuable Falcon 9 launch records.
- The Usage of EDA (Exploratory Data Analysis) as a first step of the data science project. The usage of database (SQL) to perform EDA and data visualization.
- The usage of the Interactive Visual Analytics and Dashboard module to build a Dashboard for stakeholders. Interactive visual analytics enables users to explore and manipulate data in an interactive and real-time way. Common interactions including zoom-in and zoom-out, pan, filter, search, and link. With interactive visual analytics, users could find visual patterns faster and more effectively.

Conclusions

- Analyzing launch site geolocation and proximities with Folium. Working with maps.
- Building a dashboard application with the Python Plotly Dash package. This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.
- Building a machine learning pipeline to predict if the first stage of the Falcon 9 lands successfully.
- Preprocessing allow us to standardize the data and to split our data into training and testing data, We trained the model and performed Grid Search, allowing us to find the hyperparameters that allow a given algorithm to perform best. Using the best hyperparameter values, we will determine the model with the best accuracy using the training data. You will test Logistic Regression, Support Vector machines, Decision Tree Classifier, and K-nearest neighbors. Finally, we output the confusion matrix.
- To sum up, it was found that Logistic Regression, SVM and KNN methods have the best accuracy.

Appendix

- IBM Watson Studio Project.
- GitHub
- jupyter_labs_spacex_data_collection_api.ipynb
- labs-jupyter-spacex-Data wrangling.ipynb
- jupyter-labs-eda-sql-coursera_sqllite.ipynb
- %sql select distinct Launch_Site from SPACEXTBL
- %sql select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5
- %sql select sum(payload_mass__kg_) from SPACEXTBL WHERE customer = 'NASA (CRS)'

Appendix

- %sql select avg(payload_mass__kg_) from SPACEXTBL WHERE booster_version = 'F9 v1.1'
- %sql select min(DATE) from SPACEXTBL WHERE "Landing _Outcome" = 'Success (ground pad)'
- %sql select booster_version from SPACEXTBL where "Landing _Outcome" = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000
- %sql select mission_outcome, count(mission_outcome) from SPACEXTBL GROUP BY mission_outcome
- %sql select booster_version, payload_mass__kg_ from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL)
- %sql select booster_version, launch_site from SPACEXTBL where "Landing _Outcome" = 'Failure (drone ship)' and year(DATE) = 20

Appendix

- %sql select count("Landing _Outcome"), "Landing _Outcome" from SPACEXTBL

  where DATE between '2010-06-04' and '2017-03-20' group by "Landing _Outcome"

- order by count("Landing _Outcome") desc
- jupyter-labs-eda-dataviz.ipynb
- lab_jupyter_launch_site_location.ipynb
- spacex_dash_app.py
- Dataset_part2.csv
- Dataset_part3.csv
- Confusion Matrix

Appendix

- Machine Learning Methods: Logistic Regression, Support Vector Machine, Decision Tree and k nearest neighbors KNN).

Thank you!