IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<WANG YING>
<1st April 2022>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context
  - SpaceX has gained worldwide attention for a series of historic milestones. we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers
  - What factors determine the rocket land successfully?
  - Different factors determine the success rate of successful landing.
  - What conditions will ensure a successful landing?
  - Which is the first stage will land? What's the cost of a launch?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected through SpaceX API and and web scraping from Wikipedia
- Perform data wrangling
  - One-hot encoding
- 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 different models: linear regression, SVM, Decision Tree, KNN.
  - After evaluate classification models, the accuracy of Decision Tree is the highest.

Data Collection

- Describe how data sets were collected.

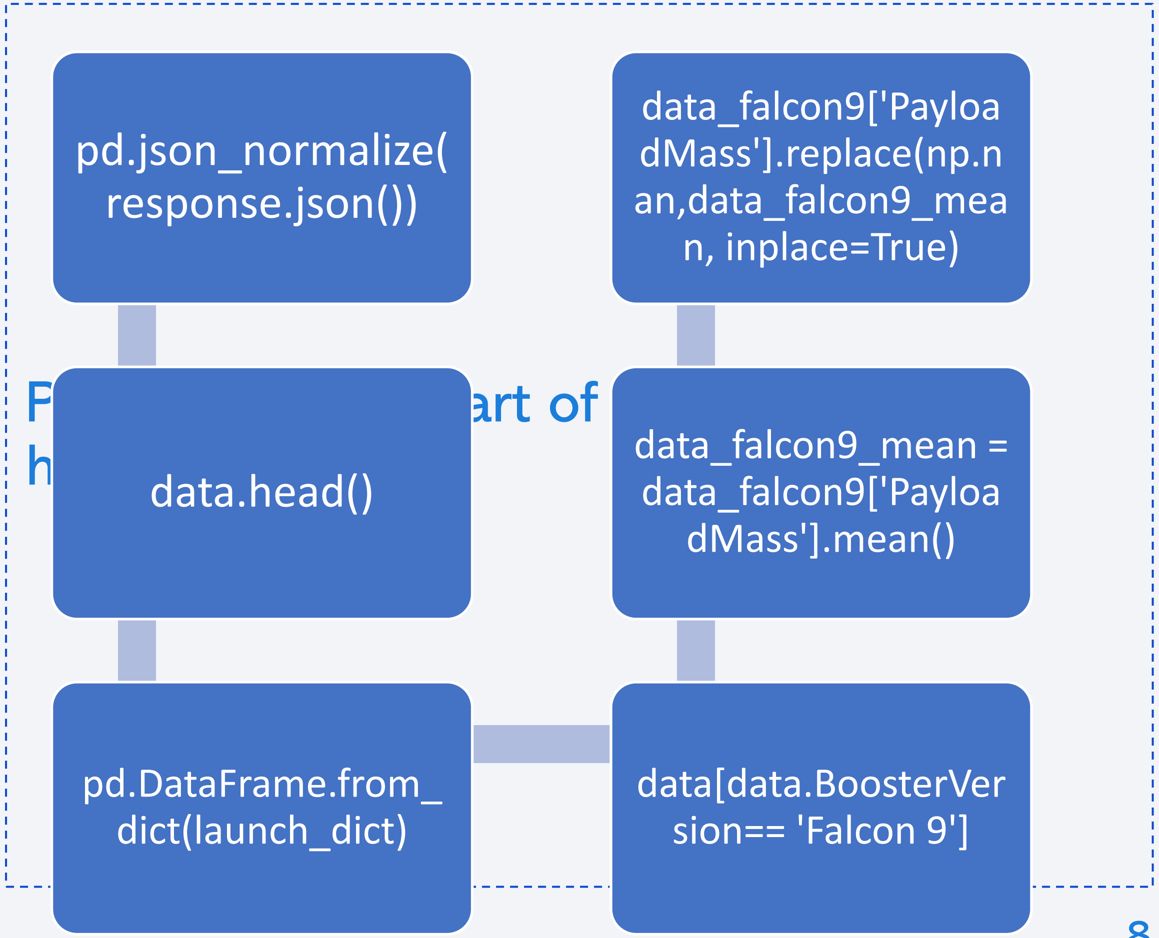
1. get request from SpaceX API and stored.
2. create a Pandas data frame .
3. clean data,check missing values and replace missing values where needed.
4. Using web scrapping from Wikipedia
5. Extract the launch records as HTML table, parse the table and convert it to a pandas dataframe

Data Collection – SpaceX API

- used the get request to the SpaceX API to collect data, clean data and data wrangling.

- GitHub URL :

- [Applied-data-science-capstone_1/C10.ipynb at master · Melodyleaf/Applied-data-science-capstone_1 (github.com)]

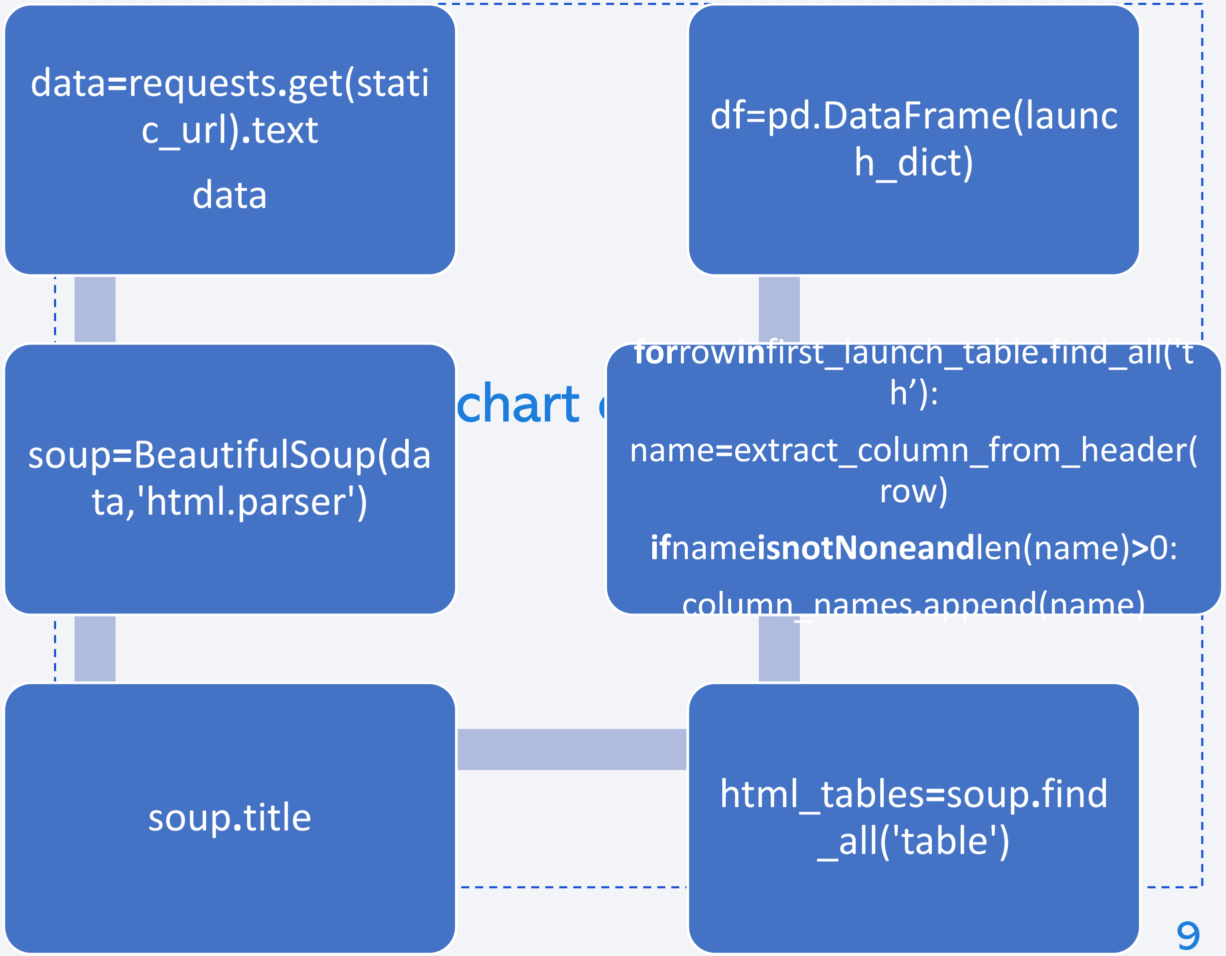


Data Collection - Scraping

- Web scraping notebook

- GitHub URL:

- [Applied-data-science-capstone_1/C10-Web Scraping.ipynb at master · Melodyleaf/Applied-data-science-capstone_1 (github.com)]()



Data Wrangling

- data wrangling process
  - Calculate the number of launches on each site
  - Calculate the number and occurrence of each orbit
  - Calculate the number and occurence of mission outcome per orbit type
  - Create a landing outcome label from Outcome column

- GitHub URL
- [Applied-data-science-capstone_1/C10-data wrangling.ipynb at master · Melodyleaf/Applied-data-science-capstone_1 (github.com)]

```
df['LaunchSite'].value_counts()
```

```
landing_class = []
for outcome in df['Outcome']:
if outcome in bad_outcomes:
landing_class.append(0);
else:
landing_class.append(1)
```

```
df['Orbit'].value_counts()
```

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

TASK 2: Calculate the number and occurrence of each orbit

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

In [6]:
```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

Out[6]:
```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

TASK 3: Calculate the number and occurence of mission outcome per orbit type

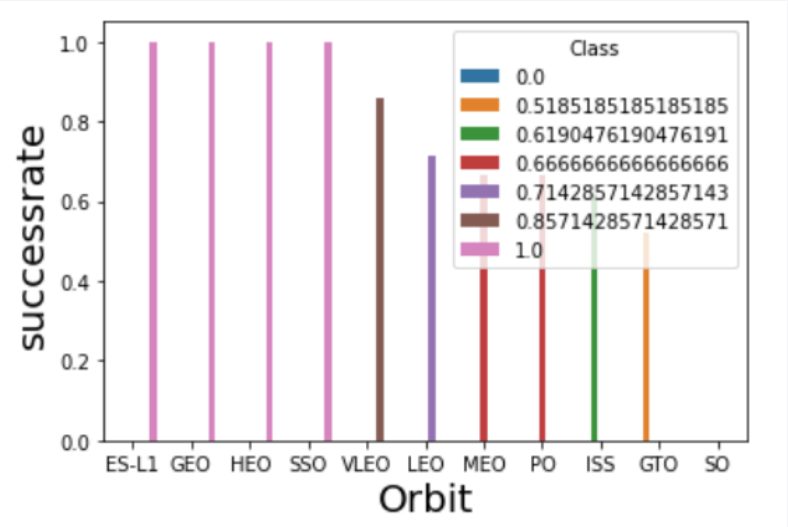
Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

In [7]:
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

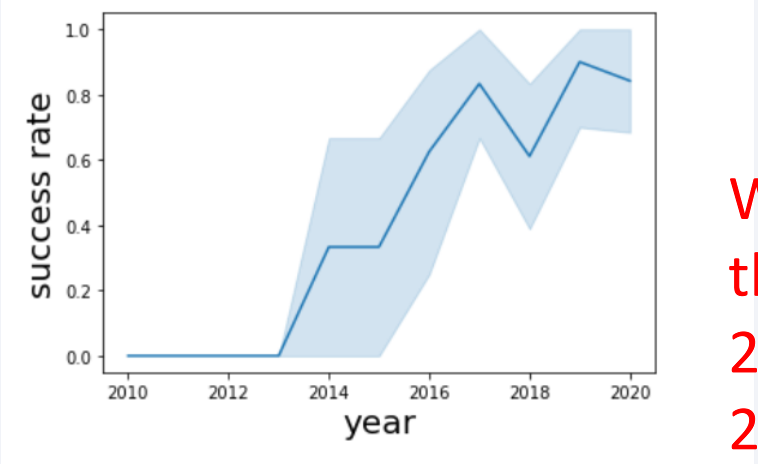
Out[7]:
```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

EDA with Data Visualization

- EDA with data visualization
- GitHub URL
- [C10-Complete the EDA with Visualization lab - IBM Cloud Pak for Data]



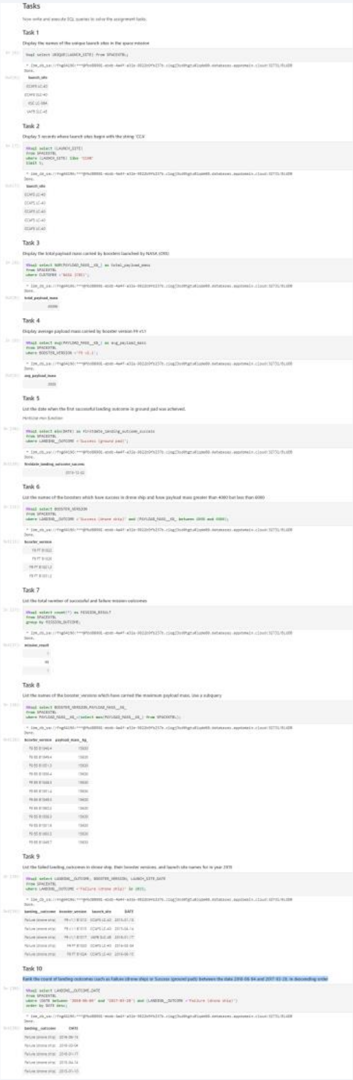
The ploted bar chart shows the first 4 orbits have high sucess rate.



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

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - 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 acheived.
  - 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
- Add the GitHub URL :
- [Applied-data-science-capstone_1/C10-Complete the EDA with SQL lab.ipynb at master · Melodyleaf/Applied-data-science-capstone_1 (github.com)]():



Build an Interactive Map with Folium

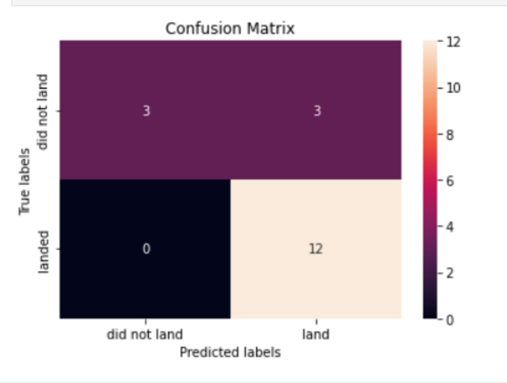
- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
  1. Mark all launch sites on a map
  2. Mark the success/failed launches for each site on the map
  3. Calculate the distances between a launch site to its proximities
- Explain why you added those objects
  - These objects tell us which launch has higher successful landing outcomes.
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
  - GithubURL:
  - Applied-data-science-capstone_1/C10-Interactive Visual Analytics with Folium lab.ipynb at master · Melodyleaf/Applied-data-science-capstone_1 (github.com)

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
  - interactive dashboard with Plotly dash was built
  - pie charts shows the total launches by a certain sites
  - scatter graph shows the relationship with Outcome and Payload Mass (Kg) of different booster version.
- Explain why you added those plots and interactions
  - These objects tell us which launch has higher successful landing outcomes.
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
  - Perform exploratory Data Analysis and determine Training Labels
  - create a column for the class/Standardize the data/Split into training data and test data
  - -Find best Hyperparameter for SVM, Classification Trees and Logistic Regression,KNN.
  - Find the method performs best using test data
- You need present your model development process using key phrases and flowchart
  - Shown as attached figure
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
  - GitHub URL:
  - Applied-data-science-capstone_1/C10-Complete the Machine Learning Prediction lab.ipynb at master · Melodyleaf/Applied-data-science-capstone_1 (github.com)



TASK 4

Create a logistic regression object then create a GridSearchCV object `logreg_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

```
In [10]:
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

```
In [11]:
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()

#create a GridSearchCV object
logreg_cv= GridSearchCV(estimator=lr,param_grid=parameters,scoring='accuracy',cv=10)
logreg_cv.fit(X_train, Y_train)
```

```
Out[11]:
GridSearchCV(cv=10, estimator=LogisticRegression(),
             param_grid={'C': [0.01, 0.1, 1], 'penalty': ['l2'],
                         'solver': ['lbfgs']},
             scoring='accuracy')
```

We output the `GridSearchCV` object for logistic regression. We display the best parameters using the data attribute `best_params\_` and the accuracy on the validation data using the data attribute `best_score\_`.

```
In [12]:
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

Results

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

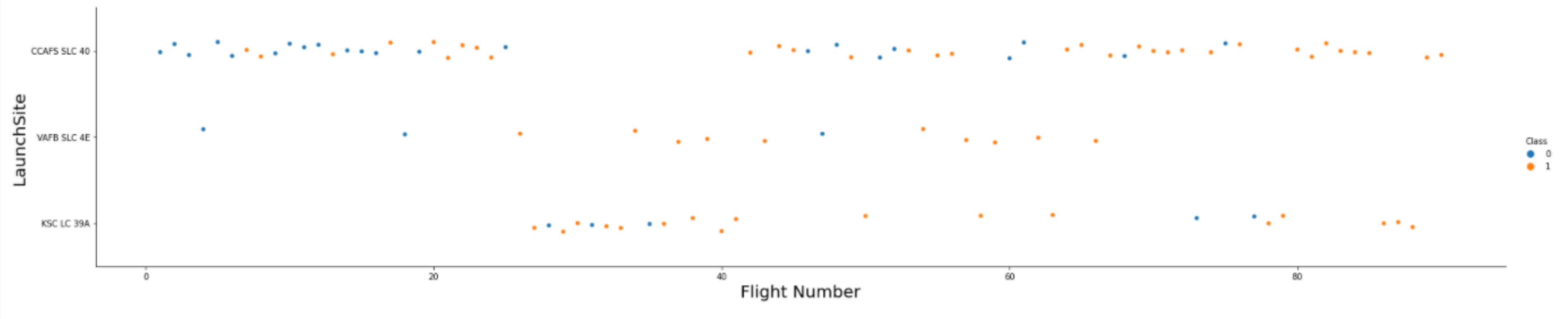
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations

Flight Number Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for higher Flight Number (greater than 80).

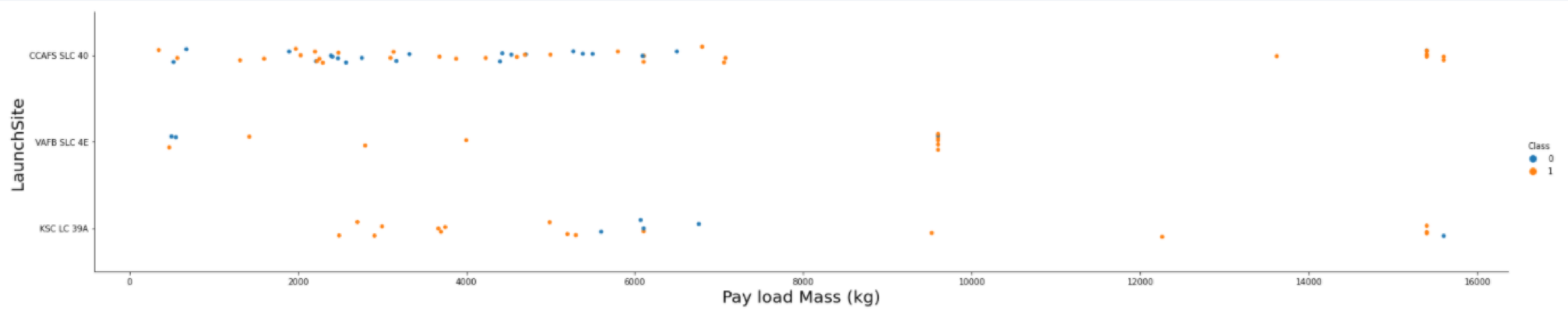


Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations

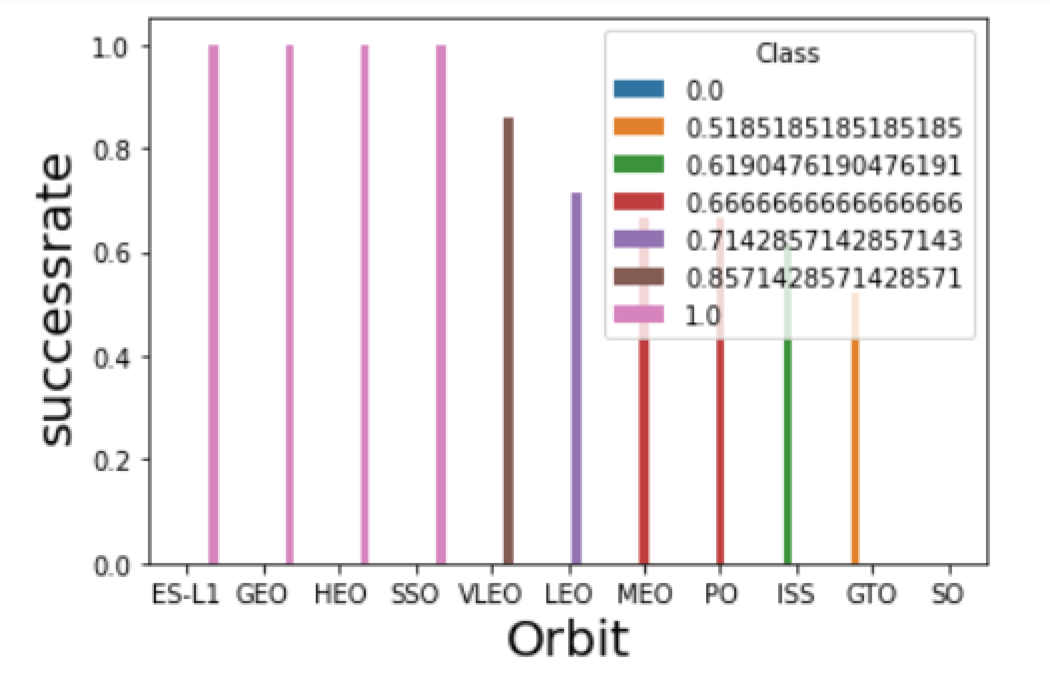
Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).



Success Rate vs. Orbit Type

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

- Show the screenshot of the scatter plot with explanations



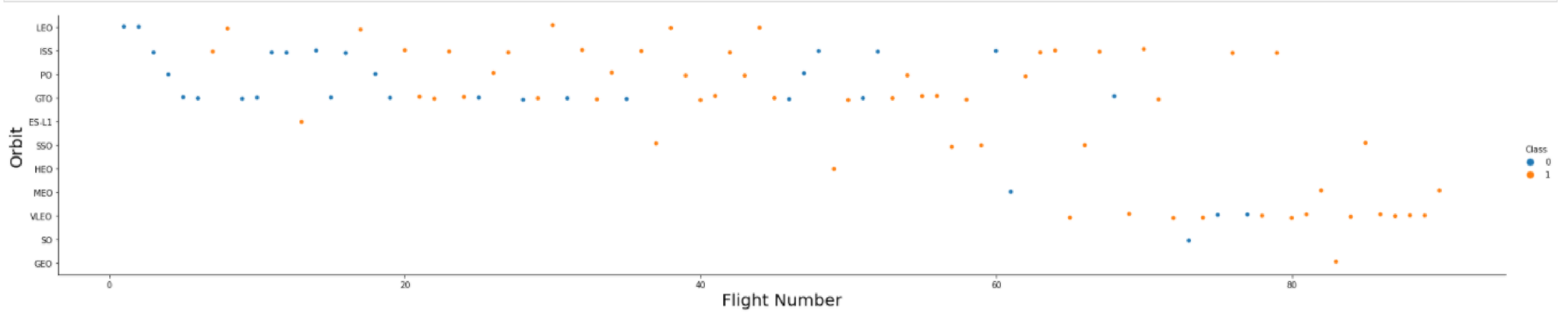
We can see that ES-L1, GEO, HEO, SSO, VLEO had the largest success rate.

Flight Number vs. Orbit Type

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

Flight Number vs. Orbit type, there is no relationship between flight number and the orbit.

- Show the screenshot of the

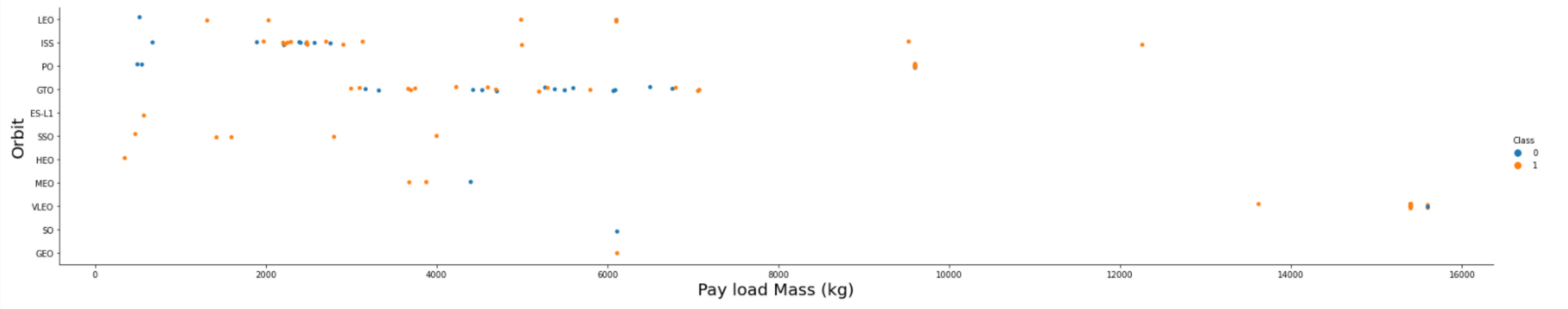


Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations

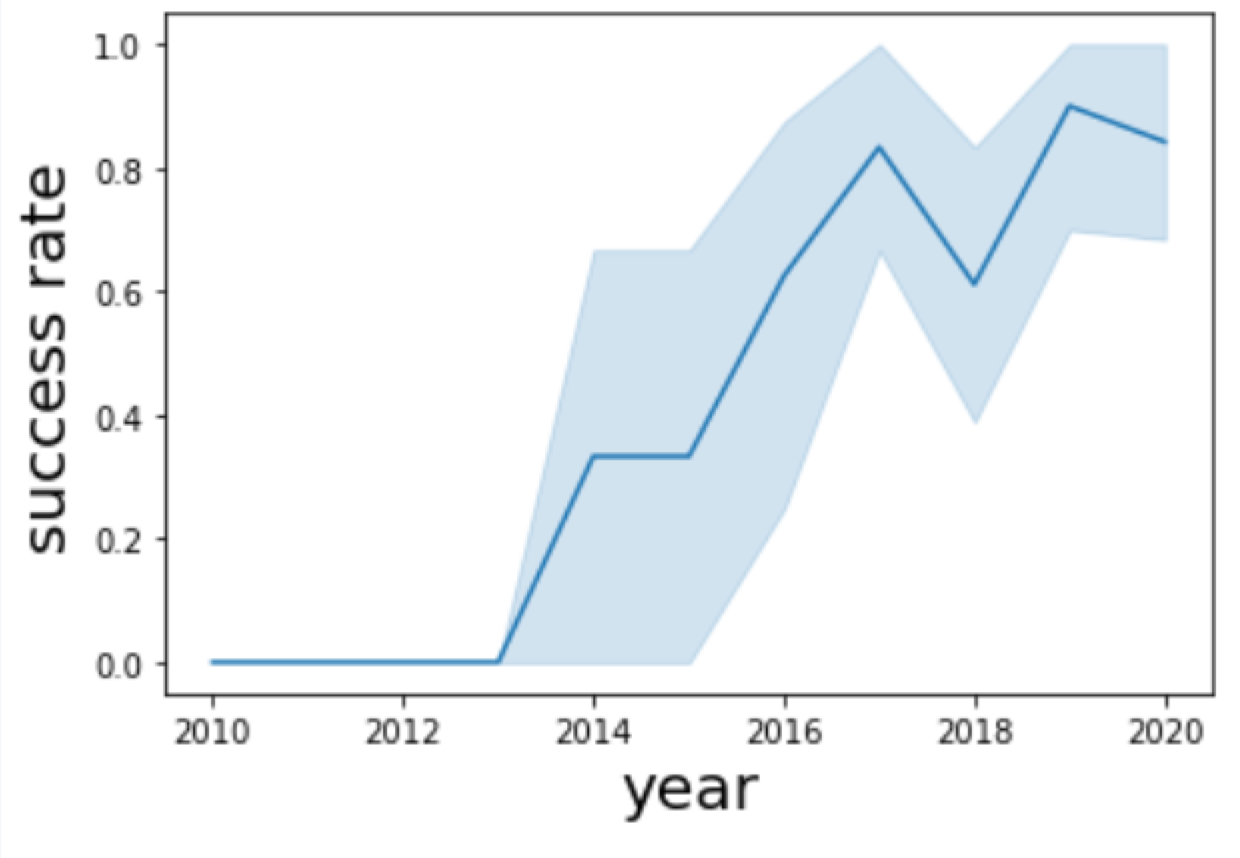
With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS. However for GTO we cannot distinguish this well as both positive landing rate and unsuccessful mission are both there here.



Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations



Sucess rate kept increasing from 2013 to 2020

All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

```
%sql select UNIQUE(LAUNCH_SITE) from SPACEXTBL;
```

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

We used the key word **UNIQUE** to show only unique launch sites from the SpaceX data.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

```
%%sql select (LAUNCH_SITE)
from SPACEXTBL
where (LAUNCH_SITE) like 'CCA%'
limit 5;
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Use key word:

Where like

Limit 5

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

```
%%sql select SUM(PAYLOAD_MASS__KG_) as total_payload_mass
from SPACEXTBL
where CUSTOMER ='NASA (CRS)';
```

total_payload_mass
45596

Use key word calculate and name it :

sum

Where

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

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

avg_payload_mass
2928

Key word:
Avg
where

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

```
%%sql select min(DATE) as firstdate_landing_outcome_success
from SPACEXTBL
where LANDING__OUTCOME ='Success (ground pad)';
```

firstdate_landing_outcome_success
2015-12-22

Use key word and name it:
Min to choose the latest calue of date
Where to choose the success gound pad

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

- Present your query result with a short explanation here

```
%%sql select BOOSTER_VERSION
from SPACEXTBL
where LANDING__OUTCOME ='Success (drone ship)' and (PAYLOAD_MASS__KG_ between 4000 and 6000);
```

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

Use key word:
Where
And: to combine two conditions
Between and: to choose mass

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

```
%%sql select count(*) as MISSION_RESULT
from SPACEXTBL
group by MISSION_OUTCOME;
```

mission_result
1
99
1

Use key word:

Group by: to group success and failure

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

```
%%sql select BOOSTER_VERSION,PAYLOAD_MASS__KG_
from SPACEXTBL
where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

Use key word:

Where

Subquery: to find the max value of mass

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

- Present your query result with a short explanation here

```
%%sql select LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE,DATE
from SPACEXTBL
where LANDING__OUTCOME ='Failure (drone ship)' in 2015;
```

landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14
Failure (drone ship)	F9 v1.1 B1017	VAFB SLC-4E	2016-01-17
Failure (drone ship)	F9 FT B1020	CCAFS LC-40	2016-03-04
Failure (drone ship)	F9 FT B1024	CCAFS LC-40	2016-06-15

Use key word:
Where:
In 2015: to choose 2015 launch
Dron_ship : to limit

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

- Present your query result with a short explanation here

```
%%sql select LANDING__OUTCOME,DATE
from SPACEXTBL
where (DATE between '2010-06-04' and '2017-03-20') and (LANDING__OUTCOME ='Failure (drone ship)')
order by DATE desc;
```

landing__outcome	DATE
Failure (drone ship)	2016-06-15
Failure (drone ship)	2016-03-04
Failure (drone ship)	2016-01-17
Failure (drone ship)	2015-04-14
Failure (drone ship)	2015-01-10

Use key word:

Where: to filter for landing outcomes

between and: to choose 2010-06-04 to 2010-03-20.

Order by: to make all outcomes in order

Desc: results in descending order.

Section 3

Launch Sites Proximities Analysis

<all launch sites on a map>

- Replace <Folium map screenshot 1> title with an appropriate title
  - all launch sites on a map
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
  - Shown as rightside figure
- Explain the important elements and findings on the screenshot
  - Map shows clearly where all launch sites are distributed, normally along the coasts.

<Mark the success/failed launches for each site on the map>

	Launch Site	Lat	Long	class	marker_color
46	KSC LC-39A	28.573255	-80.646895	1	green
47	KSC LC-39A	28.573255	-80.646895	1	green
48	KSC LC-39A	28.573255	-80.646895	1	green
49	CCAFS SLC-40	28.563197	-80.576820	1	green
50	CCAFS SLC-40	28.563197	-80.576820	1	green
51	CCAFS SLC-40	28.563197	-80.576820	0	red
52	CCAFS SLC-40	28.563197	-80.576820	0	red
53	CCAFS SLC-40	28.563197	-80.576820	0	red
54	CCAFS SLC-40	28.563197	-80.576820	1	green
55	CCAFS SLC-40	28.563197	-80.576820	0	red

- Replace <Folium map screenshot 2> title with an appropriate title
    - Mark the success/failed launches for each site on the map
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

  Shown as right figures

- Explain the important elements and findings on the screenshot
    - Red lable is launch_outcome ==0,fail.
    - Green lable is launch_outcome ==1,success.

<Calculate the distances between a launch site to its proximities>

- Replace <Folium map screenshot 3> title with an appropriate title
  - Calculate the distances between a launch site to its proximities
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
  - Shown as right figures and screen shots
- Explain the important elements and findings on the screenshot
  - The distance between launch_site and coastline,
  - Calaulated diatance is 0.48km,
  - Draw a PolyLine between a launch site to the selected coastline point


```
coastline_lat=28.56367
coastline_lon=-80.57163
launch_site_lat=28.56281
launch_site_lon=-80.57645
distance_coastline = calculate_distance(launch_site_lat, launch_site_lon, coastline_lat, coastline_lon)
print(distance_coastline,'km')
```

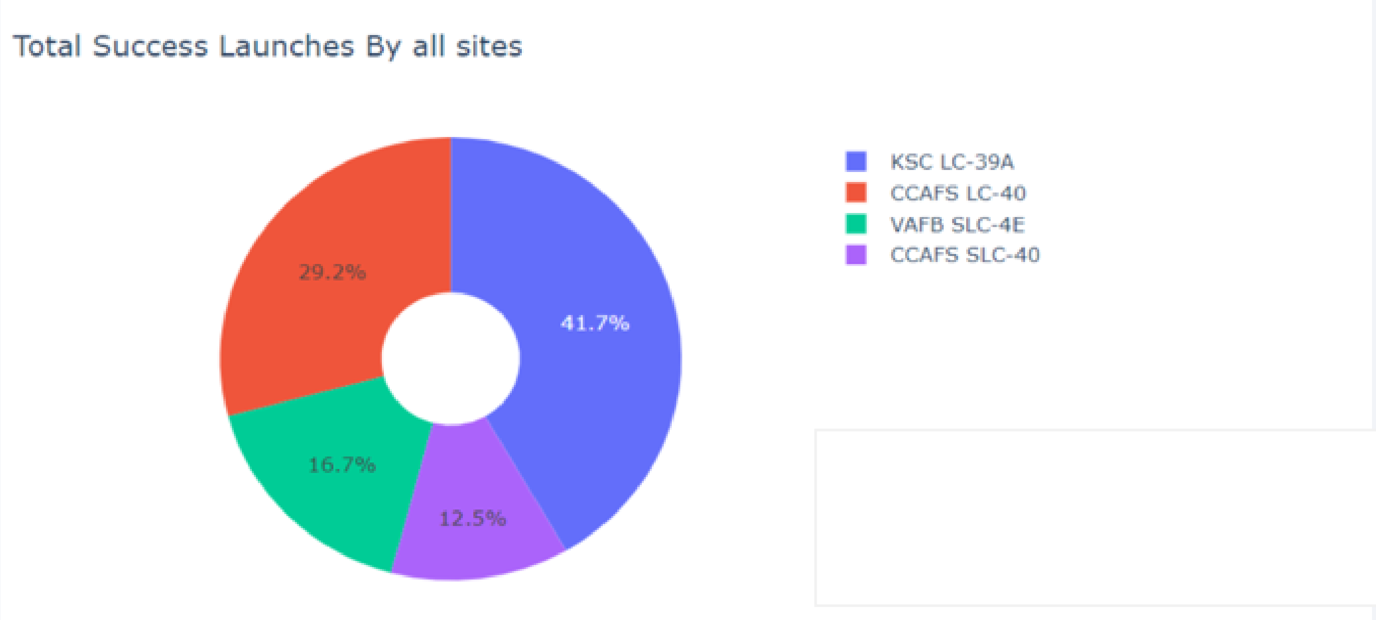
```
0.4804937589995455 km
```

Section 4

Build a Dashboard with Plotly Dash

<Pie chart showing the success percentage achieved by each launch site>

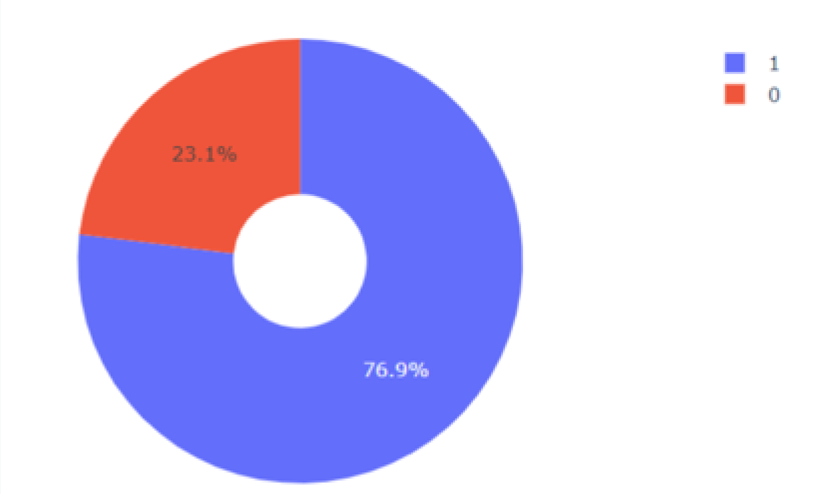
- Replace <Dashboard screenshot 1> title with an appropriate title
  - Pie chart showing the success percentage achieved by each launch site
- Show the screenshot of launch success count for all sites, in a piechart
  - Shown as below figures
- Explain the important elements and findings on the screenshot



The dark purple zone :
KSC LC-39A has the largest successful landing outcomes.

<Pie chart showing the Launch site with the highest launch success ratio>

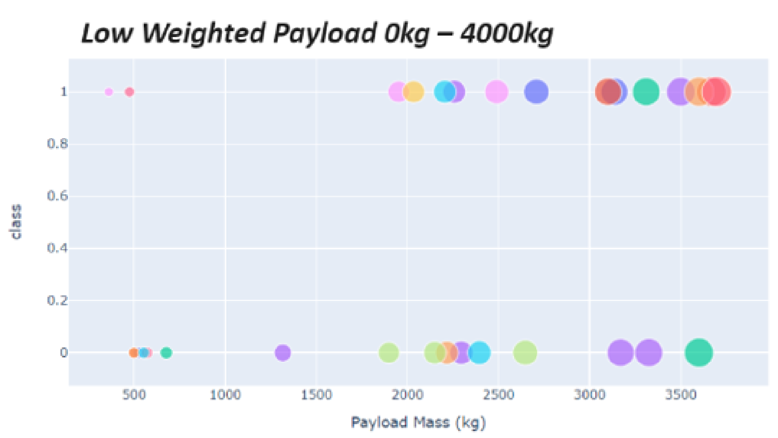
- Replace <Dashboard screenshot 2> title with an appropriate title
    - Pie chart showing the Launch site with the highest launch success ratio
- Show the screenshot of the piechart for the launch site with highest launch success ratio
    - Shown as below figure
- Explain the important elements and findings on the screenshot

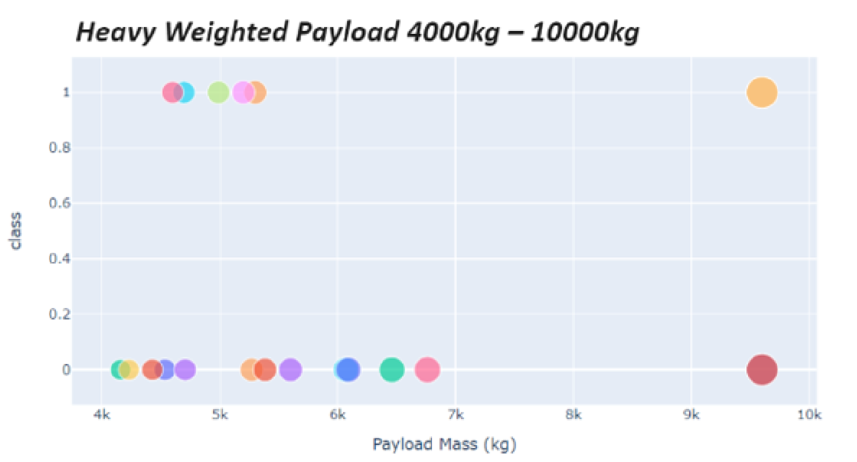


The KSC LC39A has 76.9% successful landing and 23.1% failures.

<Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider >

- Replace <Dashboard screenshot 3> title with an appropriate title
  - Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
  - Shown as below figures
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.





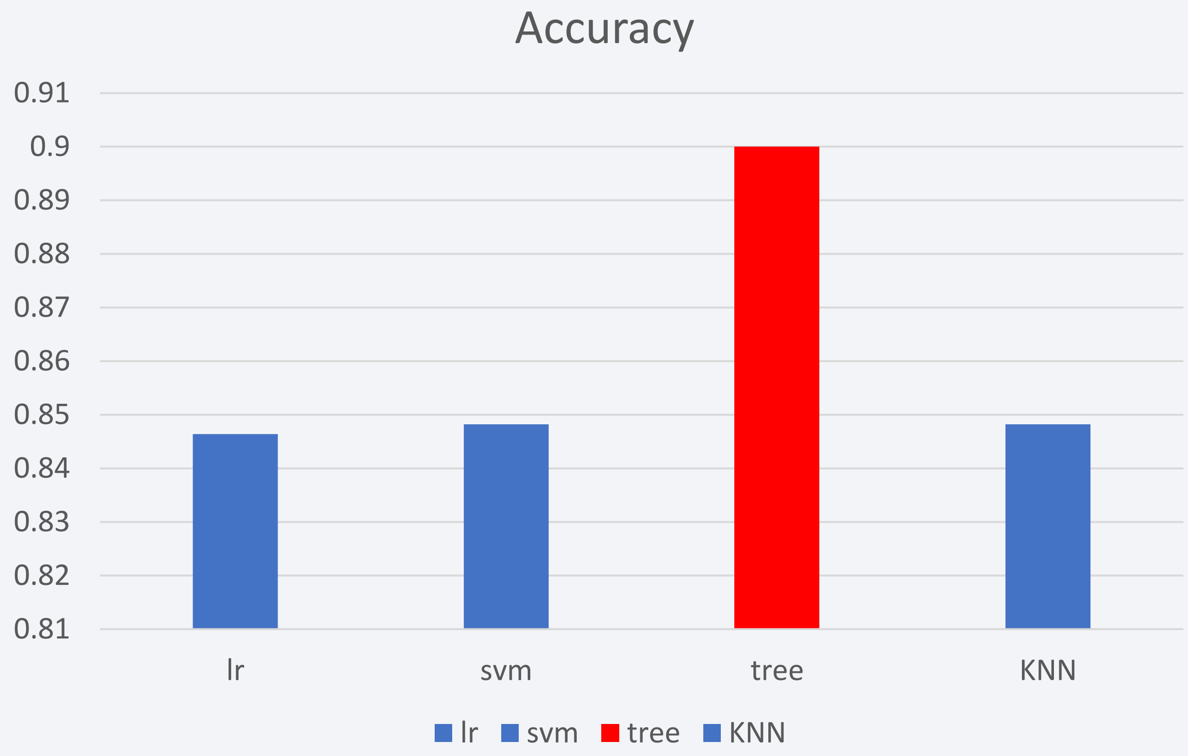
The low weighted payload (0-4000kg) has higher success rate.
The heavy weighted payload(4000-10000kg) has lower success rate.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart
  - Shown as right figures
- Find which model has the highest classification accuracy
  - Decision tree has the highest accuracy

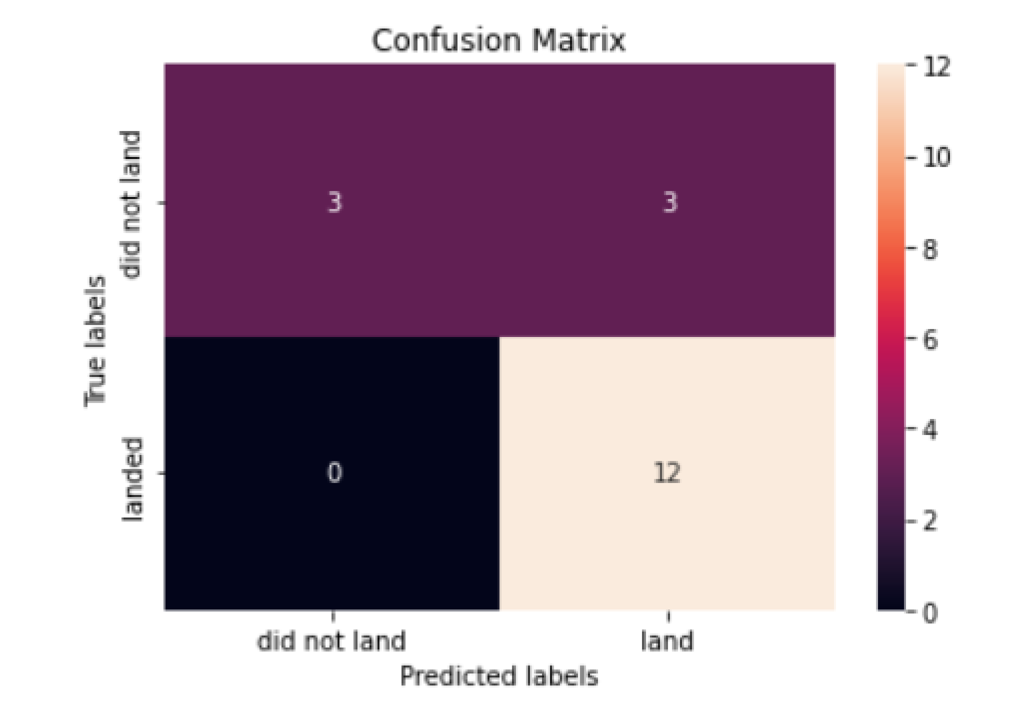


[Github URL: Applied-data-science-capstone_1/C10-Complete the Machine Learning Prediction lab.ipynb at master · Melodyleaf/Applied-data-science-capstone_1 (github.com)]

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



The confusion matrix for the decision tree shows that it can distinguish between the different landing outcomes. All successful landing have been classified.

The problem is the false positives .i.e., 3 unsuccessful landing is marked as successful landing by the classifier.

Conclusions

- Point 1 : The larger the flight number at a launch site, the greater the success rate at a launch site.
- Point 2 : Launch success rate keeps increase from 2013 to 2020.
- Point 3 : Orbits ES-L1, GEO, HEO, SSO, VLEO had the most higher success rate.
- Point 4 : KSC LC-39A has the largest success percentage achieved by each launch site.
- Point 5: The low weighted payload has higher success rate and heavy weighted payload has lower success rate.
- Point 6: The Decision tree classifier is the best machine learning algorithm due to its highest accuracy.
- Point 7: The launch sites are all not far away from the coasts.
- Point 8: For VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).
- Point 9: 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 unsuccessful mission are both there here.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!