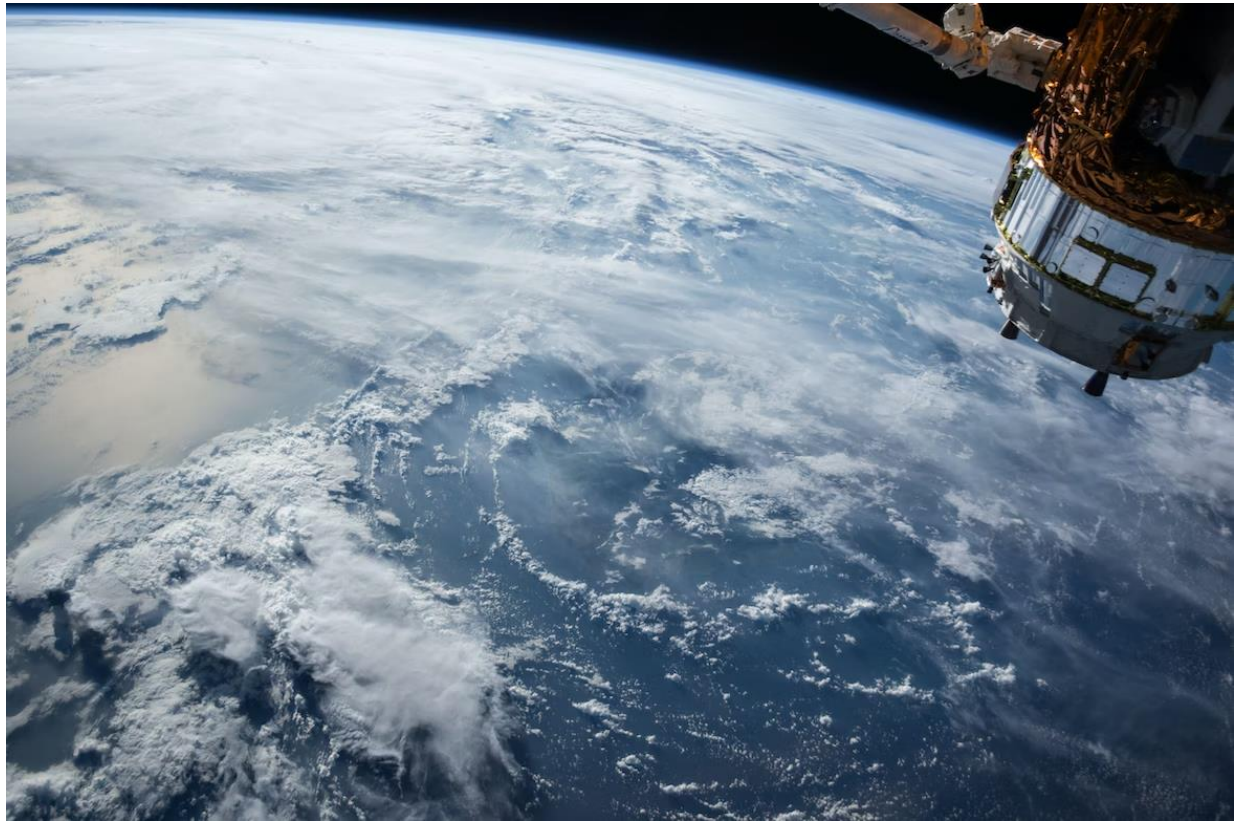
Applied Data Science Capstone: Examining SpaceX Data

Anita Wong

21 July 2023

OUTLINE



- Executive Summary
- Introduction
- Methodology
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis
  - Interactive Visual Analytics
  - Predictive Analysis
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion

EXECUTIVE SUMMARY

- Seeking to determine the price of each launch for new space company, SpaceY
- In order to compete with SpaceX, historical data of their launches was gathered and examined
  - Such data included launches, success/failure of launches, launch sites, and whether or not Stage 1 of launches were successfully able to be reused
  - In determining successes/failures of launches for SpaceX, price of launches for SpaceY can be predicted
  - Additionally, such information is used to predict if SpaceX will reuse Stage 1
- Several exploratory data analysis methods were applied to the data gathered to gain further insight
- Machine Learning also implemented for predictive analysis of future launch successes for SpaceX

INTRODUCTION

SpaceY is a future rocket company looking to compete with SpaceX in rocket launches. Through analyzing historical data from SpaceX, we were able to gather insight into how likely a launch would be successful and whether or not they would be able to reuse Stage 1. SpaceX prides itself on being an affordable rocket company since they are able to reuse the Stage 1 of their rocket launch to bring down costs.

Several exploratory data analysis methods were employed to examine the data. Several machine learning methods were applied as well to gain predictive information on future launches. An interactive web interface was also created to easily compare previous launches and launch sites with each other.

Data Collection

- BeautifulSoup, Pandas, Requests
  - These packages were used to scrape the web of the information needed in order to build the data frame that would then be cleaned and wrangled as described in the next slide
  - The information was collected from a web page after the web page was loaded into the Jupyter Notebook. An example of the code and output is as follows:
  - The following image shows the headers of all of information that would be gathered into the data frame needed to run the analyses described later in this presentation.

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

You should able to see the columns names embedded in the table header elements `<th>` as follows:

```
<tr>
<th scope="col">Flight No.
</th>
<th scope="col">Date and<br/>time (<a href="/wiki/Coordinated_Universal_Time" title="Coordinated
Universal Time">UTC</a>)
</th>
<th scope="col"><a href="/wiki/List_of_Falcon_9_first-stage_boosters" title="List of Falcon 9 first-
stage boosters">Version,<br/>Booster</a> <sup class="reference" id="cite_ref-booster_11-0"><a
href="#cite_note-booster-11">[b]</a></sup>
</th>
<th scope="col">Launch site
</th>
<th scope="col">Payload<sup class="reference" id="cite_ref-Dragon_12-0"><a href="#cite_note-Dragon-
12">[c]</a></sup>
</th>
<th scope="col">Payload mass
</th>
<th scope="col">Orbit
</th>
<th scope="col">Customer
</th>
<th scope="col">Launch<br/>outcome
</th>
<th scope="col"><a href="/wiki/Falcon_9_first-stage_landing_tests" title="Falcon 9 first-stage landing
tests">Booster<br/>landing</a>
</th></tr>
```

Data Wrangling

- Pandas & NumPy
  - Information and data such as launch sites, dates, successes/failures were collected from various websites and then sorted to create a more organized and cohesive dataset that would be easier to work with
  - The data collected was transformed and stored in a new dataset using Pandas that contained pertinent information for the future data exploration
    - This information included thing such as: Launch Site, Outcome, Booster Version, Longitude and Latitude of sites, Pay Load Mass
  - Furthering the filtration of the data, only launches of Falcon 9 was used and launches of Falcon 1 were removed
  - NumPy was used to calculate the mean of Pay Load Mass in order to replace any missing values

```
# Calculate the mean value of PayloadMass column
mean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, mean, inplace=True)
```

Exploratory Data Analysis Methodology

- SQL
  - SQL was used for the Exploratory Data Analysis (EDA) in determining several factors such as unique launch locations, and averages of payload mass. This was done to gain some initial insight in the data frame that was created from the information scraped from the web.
  - Some code used for the EDA are as follows:

```
%%sql
SELECT booster_version,PAYLOAD_MASS__KG_,"Landing_Outcome" from SPACEXTBL
where  "Landing_Outcome"='Success (drone ship)' and PAYLOAD_MASS__KG_ >4000 and PAYLOAD_MASS__KG_ < 6000
```

```
%%sql SELECT substr(Date,4,2) as month, booster_version, "Landing_Outcome" from SPACEXTBL
where "Landing_Outcome" = 'Failure (drone ship)' and substr(Date,7,4)='2015'
```

Exploratory Data Analysis Results

- From the SQL EDA applied to the data frame, some insight we gained was the Landing Outcomes of launches that occurred between 2010 and 2017:
- We can see that there were 20 successes of Stage 1 for the launch, which is a good sign and can direct further analysis in order to determine whether future launches would also be successful.

```
%%sql SELECT "Landing_Outcome", COUNT(*) as 'Quantity' FROM SPACEXTBL
WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017' GROUP BY "Landing_Outcome" ORDER BY "Quantity"
```

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

Landing_Outcome	Quantity
No attempt	1
Controlled (ocean)	2
Failure (parachute)	2
Failure	3
Failure (drone ship)	3
Success (ground pad)	7
Success (drone ship)	8
No attempt	10
Success	20

Exploratory Data Analysis Results

- We also found the average of the Pay Load Mass (in kilograms) that was carried by the booster version F9 v1.1:

Task 4

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

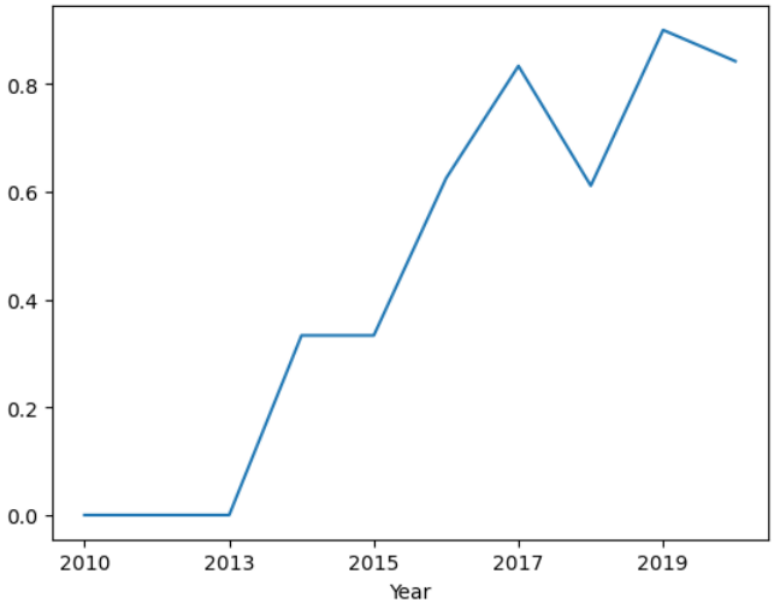
Interactive Visual Analytics Methodology

- Seaborn, MatPlotLib, Folium, Plotly
  - These packages were used to give us more insight visually on how each launch site performed and allowed us to compare several variables against each other.
  - Some comparisons included Success Rate against Orbit, Success Rate against Year, and visualizing payload mass by flight number.
  - Folium specifically was used to create maps that documented all of the launch sites to show us how they compared against each other, also giving us a comparison to NASA.
  - Plotly was utilized to create an interactive dash that allowed for each comparison of each launch and could be adjusted to reflect information for a specific launch

Interactive Visual Analytics Results

- One result pulled was creating a line chart to visual the rate of success per year. From the output to the right, we see that successful launches grew steadily up until 2017 when there was a slight dip before successful launches occurred again.

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
temp_df = df.copy()
temp_df['Year'] = year
temp_df.groupby('Year')['Class'].mean().plot()
sns.lineplot(data=temp_df, x=np.unique(Extract_year(df['Date'])), y=temp_df.groupby('year')['Class'].mean())
plt.xlabel("Year",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```

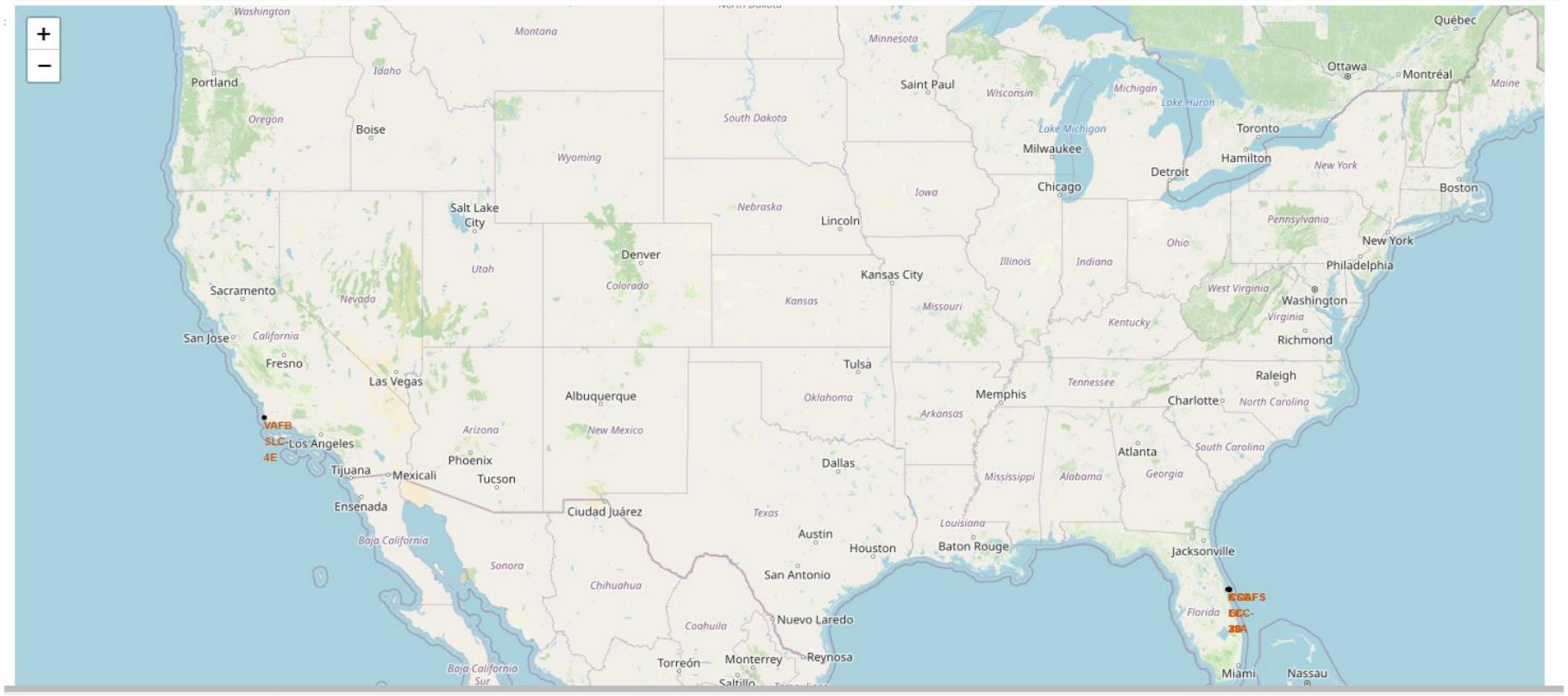


Interactive Visual Analytics Results: Folium Map

- The below image and code allowed us to view the difference in launch sites. We see that launches were conducted on both coasts of the United States of America (indicated by the black dots and red labels).

```
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
# For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label

for index, row in launch_sites_df.iterrows():
    coordinate = [row['Lat'], row['Long']]
    folium.Circle(coordinate, radius=1000, color='#000000', fill=True).add_child(folium.Popup(row['Launch Site'])).add_to(site_map)
    folium.map.Marker(coordinate, icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0), html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % row['Launch Site'], )).add_to(site_map)
site_map
```

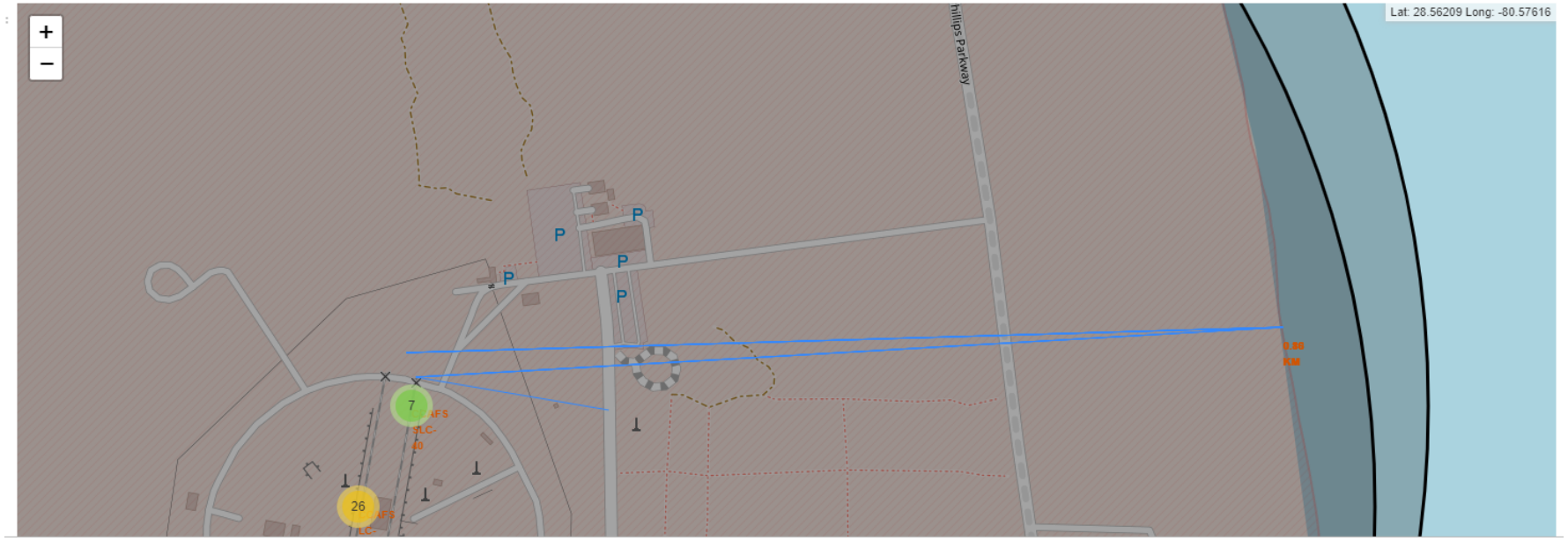


Interactive Visual Analytics Results: Folium Map

- The below code and output provided us with a Folium map that demonstrated how far away a launch site was from the coast as well as the closest road. This information is important to determine whether or not a roadway would impede on the launch.

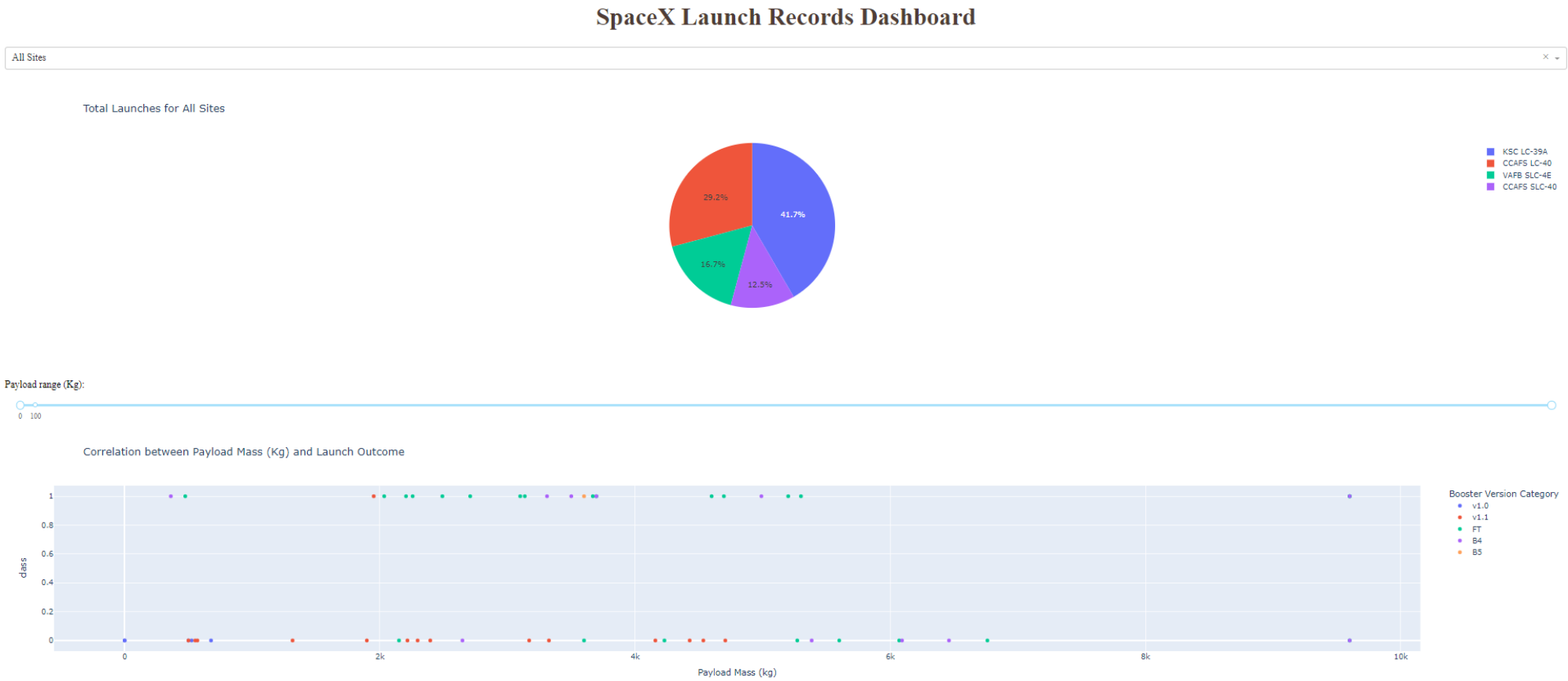
```
coordinates = [
    [28.56345, -80.57678],
    [28.56316, -80.57484]]

lines = folium.PolyLine(locations=coordinates, weight=1)
site_map.add_child(lines)
distance = calculate_distance(coordinates[0][0], coordinates[0][1], coordinates[1][0], coordinates[1][1])
distance_marker = folium.Marker(
    [28.5638, -80.56804],
    icon=DivIcon(
        icon_size=(20,20),
        icon_anchor=(0,0),
        html='<div style="font-size:12; color:#d35400;"><b>%s</div>' %"{:10.2f} KM".format(distance),
    )
)
site_map.add_child(distance_marker)
site_map
```



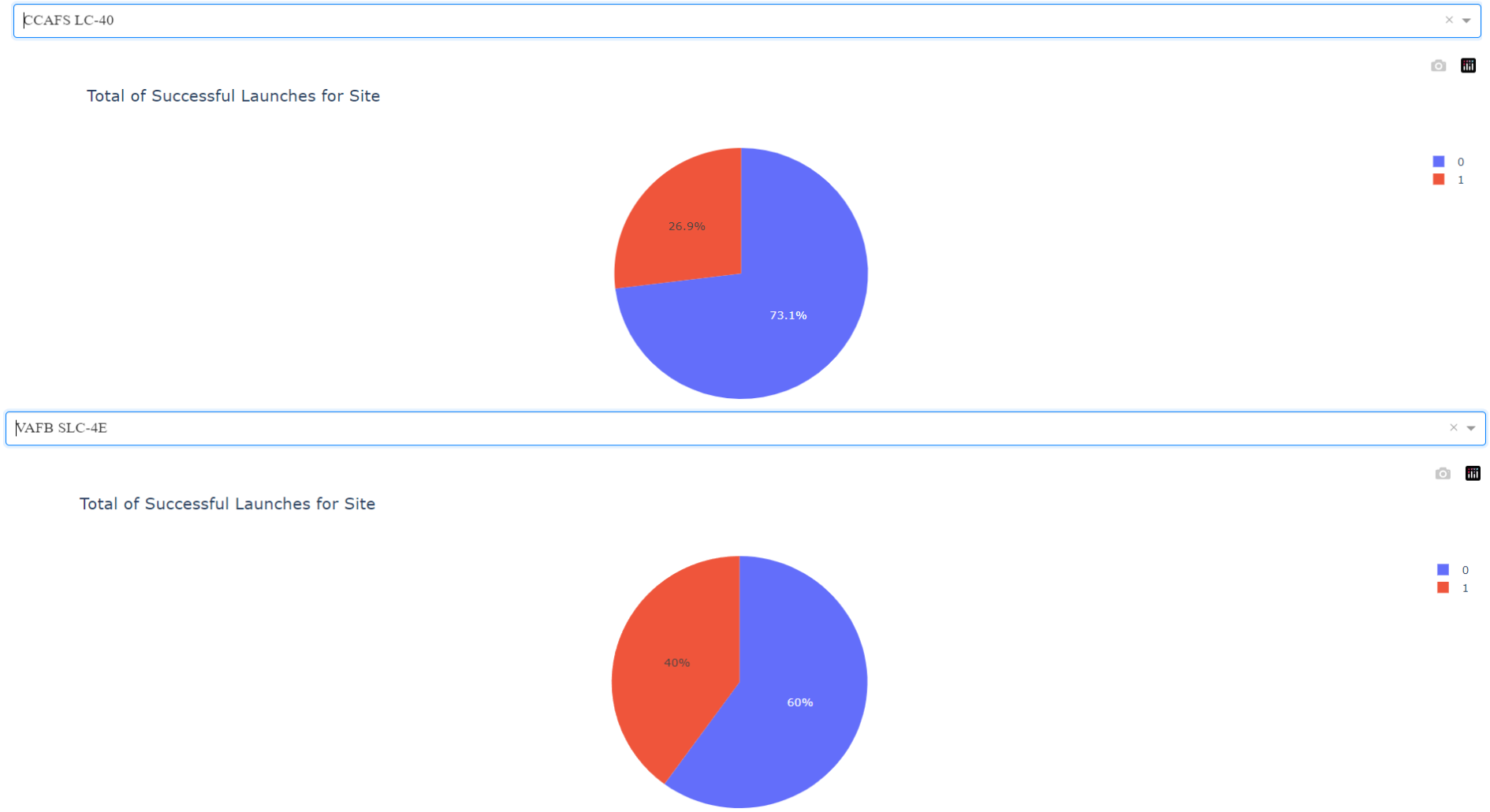
Interactive Visual Analytics: Plotly Dash

- The Plotly Dash was created in order to compare the launch sites against each other.



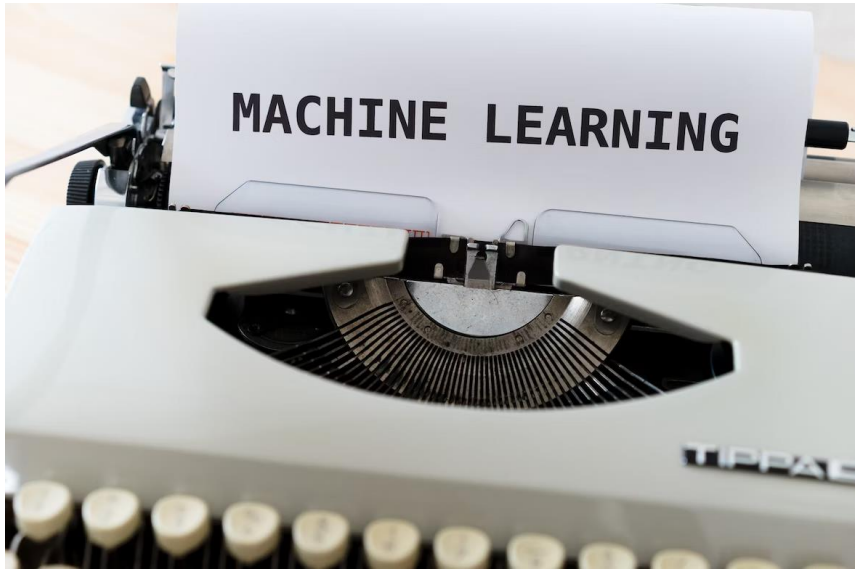
Interactive Visual Analysis: Plotly Dash

- We can compare the total successes for two different launch sites as seen below:



Predictive Analysis Methodology:

- NumPy, Pandas, SciKitLearn
  - We utilized several predictive analysis methods and packages in order to predict the outcome of future launches. The methods include: Logistic Regression, Support Vector Machine (svm), Decision Tree Classifier, and K Nearest Neighbors. Confusion matrixes were made for all four methods to determine the accuracy from test data after they were trained using a training data set.



Predictive Analysis Results: Logistic Regression

- Below is the accuracy result and confusion matrix for Logistic Regression after it was trained with the training data set. We see that it scored .8333 in accuracy with the testing data set, but had quite a bit of a problem with false positives:

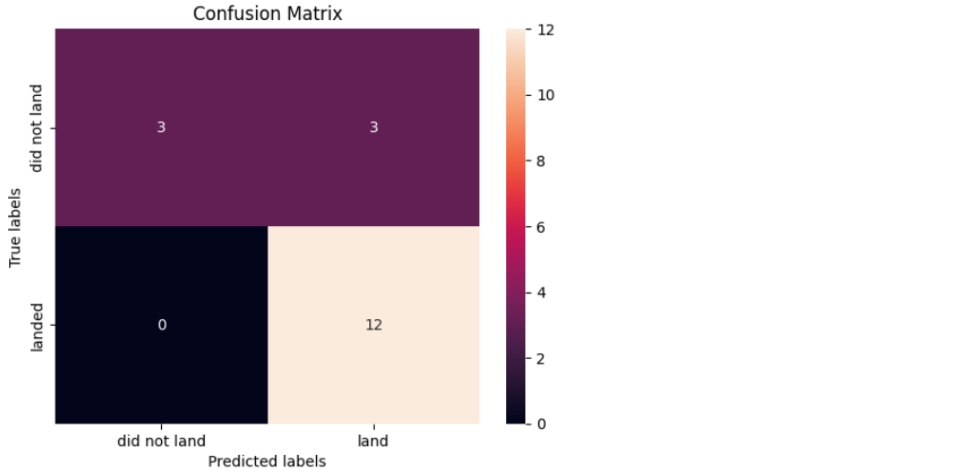
Calculate the accuracy on the test data using the method `score`:

In [24]:
```
logreg_cv.score(X_test, Y_test)
```

Out[24]:
```
0.8333333333333334
```

Lets look at the confusion matrix:

In [25]:
```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Predictive Analysis Results: SVM

- Below is the accuracy and confusion matrix of the SVM:
- We can see that it is quite similar to the Logistic Regression.

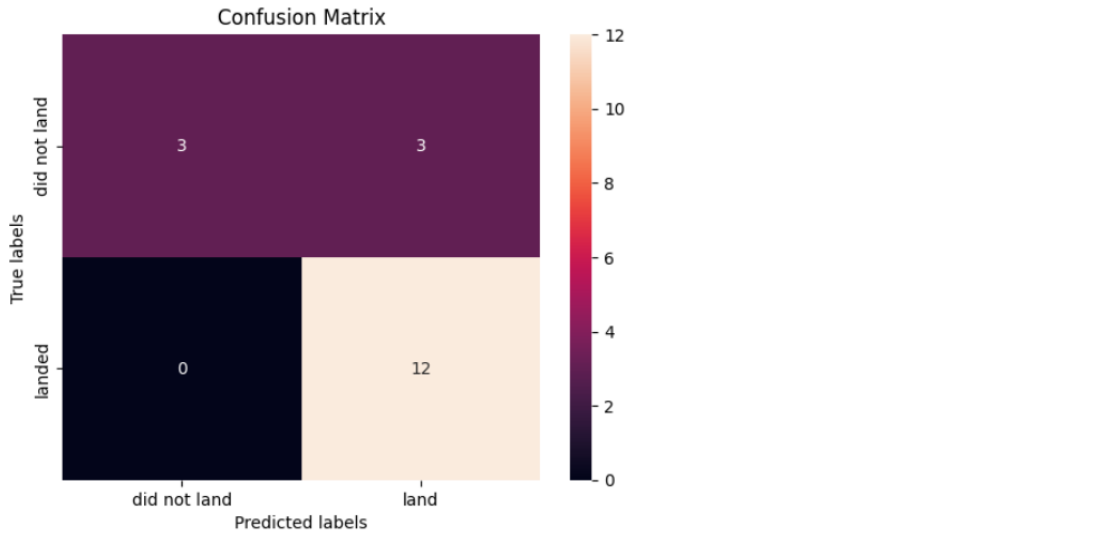
Calculate the accuracy on the test data using the method `score`:

```
svm_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

We can plot the confusion matrix

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



Predictive Analysis Results: Decision Tree

- Below is the accuracy and confusion matrix for the Decision Tree that was run against the training and test data sets:
- Again, we see that the score and matrix are similar to the previous ones seen for Logisitic Regression and SVM.

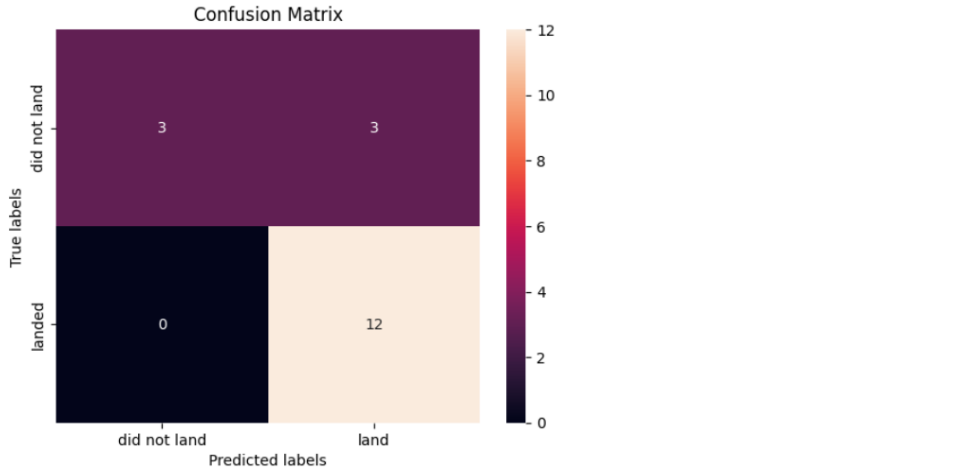
Calculate the accuracy of tree_cv on the test data using the method `score`:

```
tree_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

We can plot the confusion matrix

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



Predictive Analysis Results: K Nearest Neighbors

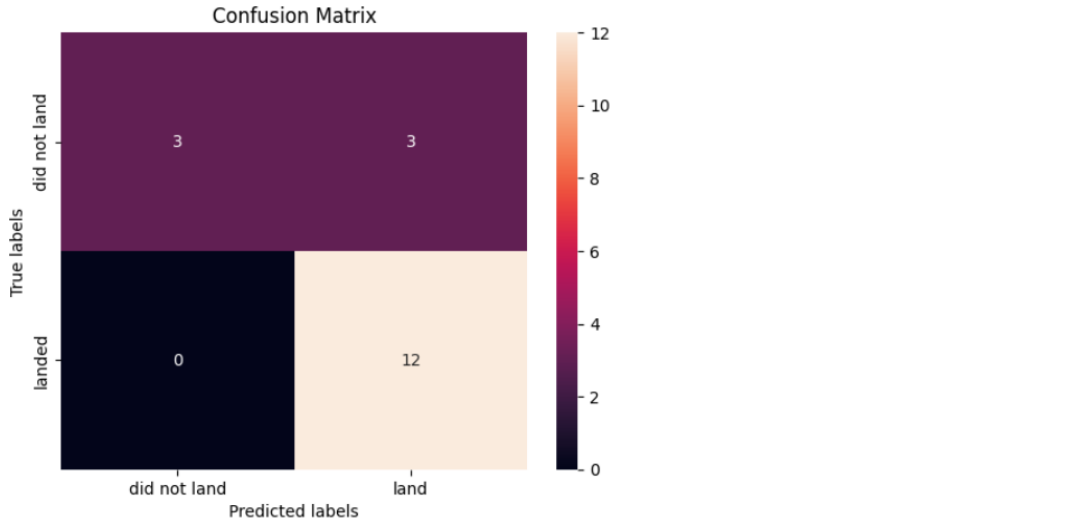
- Lastly, we have the accuracy and confusion matrix of the K Nearest Neighbors:
- We see once again that it yielded an accuracy of .833.

```
knn_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

We can plot the confusion matrix

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



Discussion & Implications

Upon reviewing the results from the Predictive Analysis, we see that all of the values of accuracy are the same. This is due to the nature of the data set that was used – since the amount was small, we calculated generally the same value despite using different techniques.

Additionally, from comparing the launch site successes on the Plotly Dash, we can see that launch sites on the East Coast yielded higher rates of success compared to the launch site on the West Coast.

Conclusion:

- Based on the findings presented, we can infer that more successes may be attributed to launching on the West Coast so this may influence the creation of future launch sites.
- Additionally, more data should be collected from launches in order to expand the data set so that we can gain more insight. From the predictive analyses conducted on the data set created, the accuracy scores generated were the same despite the method used. Expanding the data set with more information could give us a better idea of which predictive analysis method would work better at determining a successful launch.