IBM Data Science Capstone Project

SpaceX Falcon 9 Rocket Analysis

Charles Schultz

August 2023

Table of Contents

Executive Summary

As the aerospace industry evolves, SpaceX's innovative approach to rocket reusability has disrupted traditional norms, enabling them to significantly reduce launch costs. This project aims to empower a competing startup to make informed bids against SpaceX by accurately estimating the probability of a successful first stage landing through the continued reuse of a booster, related to SpaceX's Falcon 9 rocket.

Through the analysis of historical data, performance metrics, and factors influencing successful landings, our data science team will develop predictive models. These models will consider variables such as launch parameters and technical specifications to forecast landing outcomes.

By utilizing precise predictive models grounded in data science findings, the competing startup can strategically tailor their bids. Incorporating projected cost savings from a successful landing, the startup can submit more competitive bids in the rocket launch market.

By harnessing these results, we aspire to equip them with the necessary tools to strategically estimate the probability of successful first stage landings for Falcon 9 rocket launches. In doing so, the startup can confidently compete against SpaceX, leveraging accurate cost projections and potentially disrupting the space industry's landscape.

Introduction

SpaceX, founded by Elon Musk in 2002, has revolutionized the aerospace industry by introducing groundbreaking technology with significantly reduced production costs. Despite its youth, SpaceX has achieved over a 50% reduction in launch expenses, with further savings anticipated through its Starship project. The reusability of the first stage booster, which represents 70% of the rocket's costs, is central to this success. In 2005, SpaceX developed the Falcon 9 rocket, a reusable heavy lift vehicle which achieved its first successful landing and recovery of a first stage booster in December 2015 with Falcon 9 Flight 20.

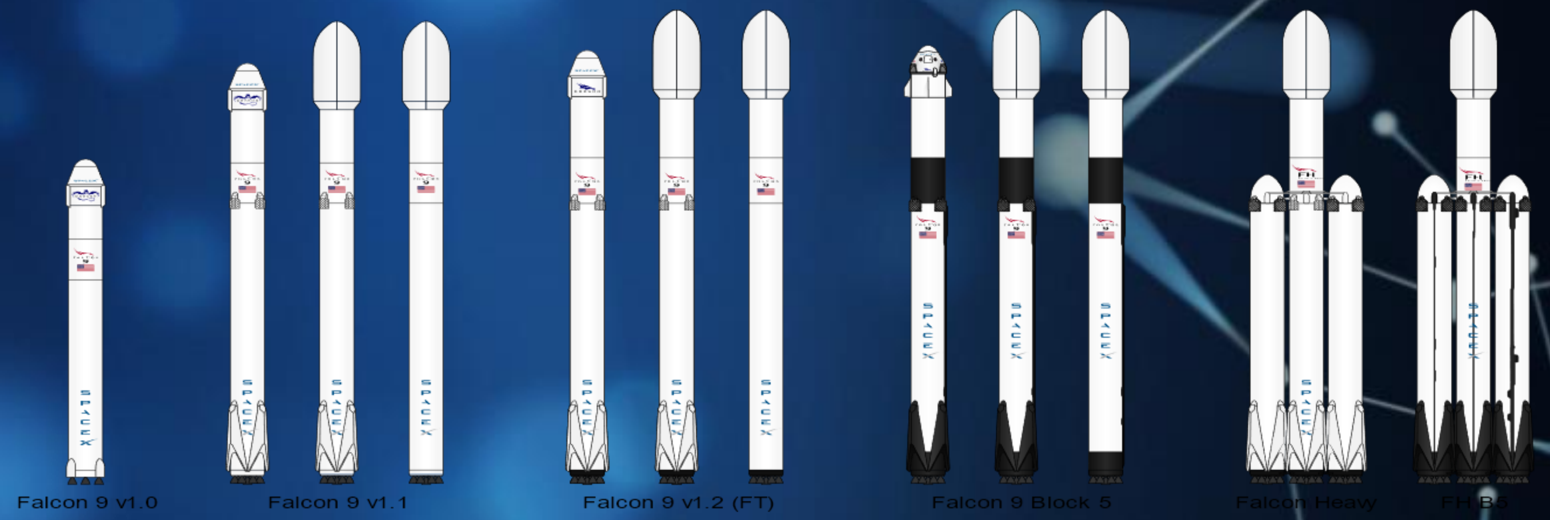
The Falcon 9 rocket holds a remarkable advantage over its competitors due to its reusability feature. With a launch cost of $62 million, significantly lower than other providers whose costs exceed $165 million per launch, SpaceX's ability to reuse the first stage booster has been pivotal in achieving this feat. Consequently, predicting the outcome of the first stage landing is key to determining the overall cost of a launch for potential competitors.

To achieve this objective, we will draw from diverse data sources, including web scraping from Wikipedia and the SpaceX API. We will analyze historical data, performance metrics, and key factors influencing landing outcomes using various data collection, analytical, and machine learning programs. Through this analysis, we will pave the way for predictive models that consider launch parameters and technical specifications to accurately forecast the probability of successful landing events.

For startup companies interested in challenging SpaceX's dominance, comprehending the likelihood of Falcon 9's first stage landing success is vital to their research. Equipped with precise predictive models grounded in robust data science insights, these startups can craft strategic bids. By factoring in the anticipated cost reductions from successful SpaceX landings, they can present exceptionally competitive proposals within the aerospace industry.

GitHub Repository URL

https://github.com/cschultz76/IBM-Data-Science-Capstone-Project.git



Summary of Methodologies

Data Collection

- GET Requests from the SpaceX REST API
- Web Scraping of SpaceX Falcon 9 Launches from Wikipedia

Data Wrangling and Cleaning

- Filtering and Data Organization through Python within IBM Watson Studio

Exploratory Data Analysis

- SQL Database Manipulation of SpaceX Datasets
- Data Enquiry through Pandas and Matplotlib Libraries within Python Jupyter Notebook

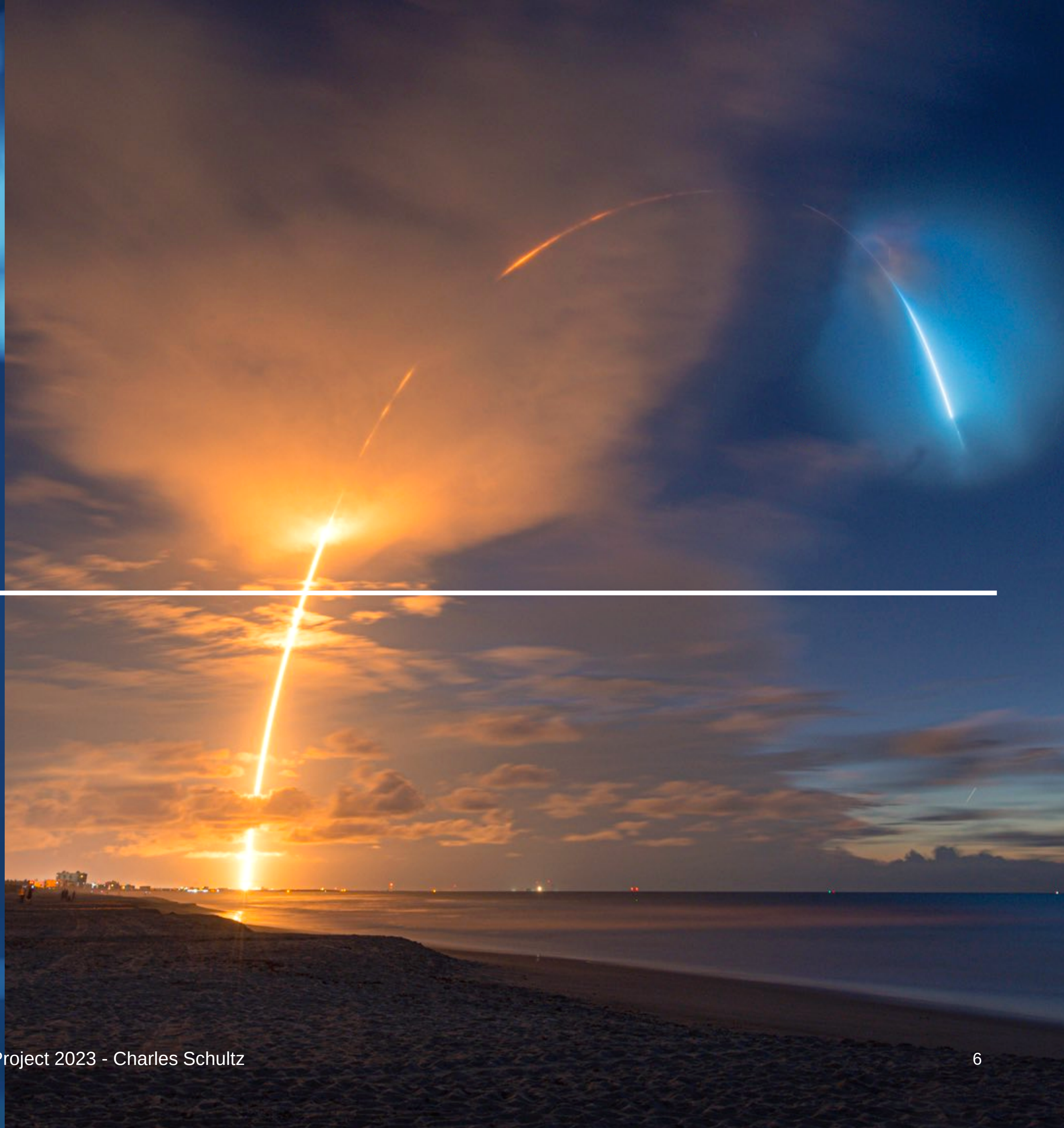
Visual Analysis

- Launch Site Geospatial Mapping through Folium
- Interactive Dashboard Launch Maps within Plotly Dash

Data Modeling

- Machine Learning for Launch Outcomes Utilizing Sci-Kit Learn Library in Python Jupyter Notebook

Methodology



Data Collection from the SpaceX API

GitHub Data Collection API Link

Utilized the SpaceX API to gather essential launch data which included booster version, launch site, orbit, and landing outcome.

- Created a GET request to the SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

- Decoded data through JSON and converted into a dataframe

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

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

- Created a dataframe subset to focus on pertinent launch information

```
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

data['date'] = pd.to_datetime(data['date_utc']).dt.date

data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

- Filtered data for Falcon 9 booster launches and converted null payload mass entries to a mean value

```
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

```
payloadmassavg = data_falcon9['PayloadMass'].mean()

data_falcon9['PayloadMass'].replace(np.nan, payloadmassavg, inplace=True)
```

Web Scraping from Wikipedia

Github Web Scraping Link

Web scraped SpaceX's page from Wikipedia to retrieve historical launch records of the Falcon 9 rocket.

- Performed a web scraping request to Wikipedia for launch data

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
data  = requests.get(static_url).text
```

```
soup = BeautifulSoup(data, 'lxml')
```

- Extracted column names from scraped information

```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

- Created an empty dictionary from extracted column names in order to create a Pandas dataframe

```
launch_dict= dict.fromkeys(column_names)

del launch_dict['Date and time ( )']

launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Web Scraping from Wikipedia - Continued

- Imported scraped launched information into the newly created library

```
extracted_row = 0
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
    for rows in table.find_all("tr"):
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        row=rows.find_all('td')
        if flag:
            extracted_row += 1
            launch_dict['Flight No.'].append(flight_number)
            datatimelist=date_time(row[0])

            date = datatimelist[0].strip(',')
            launch_dict['Date'].append(date)

            time = datatimelist[1]
            launch_dict['Time'].append(time)

            bv=booster_version(row[1])
            if not(bv):
                bv=row[1].a.string
            launch_dict['Version Booster'].append(bv)

            launch_site = row[2].a.string
            launch_dict['Launch site'].append(launch_site)

            payload = row[3].a.string
            launch_dict['Payload'].append(payload)

            payload_mass = get_mass(row[4])
            launch_dict['Payload mass'].append(payload_mass)

            orbit = row[5].a.string
            launch_dict['Orbit'].append(orbit)

            customer = row[6].text.strip()
            launch_dict['Customer'].append(customer)

            launch_outcome = list(row[7].strings)[0]
            launch_dict['Launch outcome'].append(launch_outcome)

            booster_landing = landing_status(row[8])
            launch_dict['Booster landing'].append(booster_landing)
```

Data Wrangling and Manipulation

Github Data Wrangling Link

Performed initial exploratory data analysis to compile launch site and orbit information in order to predict booster landing outcomes.

- Calculated the number of Falcon 9 launches per site and per orbit classification

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

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

- Created an orbit outcome dataframe

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

- Calculated the amount of failed launches

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

- Classified landing outcomes into a new Landing_Column

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

- Determined the success rate of overall launches

```
df["Class"].mean()
```

```
3]:  0.6666666666666666
```

Exploratory Data Analysis Visualization

Github EDA with Visualization Link

Utilized previously wrangled SpaceX data to create bar, line, and scatter plot charts to visualize various Falcon 9 launch statistics.

- Generated a scatterplot graph which classified each launch by its flight number and payload mass

```
sns.catplot(y="PayloadMass", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```

- Created a scatterplot graph which visualizes the relationship between the flight number and its launch site

```
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

- Compiled a bar chart demonstrating successful launches within each orbit type

```
df.groupby("Orbit").mean()['Class'].plot(kind='bar')
plt.xlabel("Orbit Type",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```

- Plotted a line chart which visualizes successful rocket launches by year

```
sns.lineplot(data=df, x='Date', y='Class')
plt.xlabel("Year", fontsize=20)
plt.ylabel("Success Rate", fontsize=20)
plt.show()
```

Exploratory Data Analysis with SQL

Github EDA with SQL Link

Continued EDA progress by queuing up specific Falcon 9 flight data through SQL.

- Queried the average payload carried per launch by the Falcon 9 version 1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
AVG(PAYLOAD_MASS__KG_)
2928.4
```

- Determined which launches were successful with a payload mass between 4000 and 600 kg

```
%sql SELECT PAYLOAD \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

- Calculated failed drone ship landings during 2015 based off of their booster version and launch site

```
%sql SELECT substr(Date,4,2) as month, DATE,BOOSTER_VERSION, LAUNCH_SITE, [Landing _Outcome] \
FROM SPACEXTBL \
where [Landing _Outcome] = 'Failure (drone ship)' and substr(Date,7,4)='2015';
```

- Ranked all successful and failed launches between 2010 and 2016

```
%sql SELECT [Landing _Outcome], count(*) as count_outcomes \
FROM SPACEXTBL \
WHERE DATE between '04-06-2010' and '20-03-2017' group by [Landing _Outcome] order by count_outcomes DESC;
```

Data Visualization through Folium

Github Visualization with Folium Link

Utilized compiled EDA information to produce interactive launch site maps.

- Created a new dataframe with launch site latitude and longitude information

```
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```

- Generated a new interactive map with the Nasa Johnson Space Center as the center of interest

```
nasa_coordinate = [29.559684888503615, -95.0830971930759]
site_map = folium.Map(location=nasa_coordinate, zoom_start=10)
```

- Added a map marker location for each Falcon 9 launch Site

```
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
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
```

- Created a new dataframe listing successful and failed rocket launches

```
def assign_marker_color(launch_outcome):
    if launch_outcome == 1:
        return 'green'
    else:
        return 'red'

spacex_df['marker_color'] = spacex_df['class'].apply(assign_marker_color)
spacex_df.tail(10)
```

Data Visualization through Folium - Continued

- Generated failed and successful launch site markers to the map

```
marker_cluster = MarkerCluster()
site_map.add_child(marker_cluster)
for index, row in spacex_df.iterrows():
    lat = row['Lat']
    long = row['Long']
    launch_site = row['Launch Site']
    launch_marker = folium.Marker(
        [lat, long],
        icon=folium.Icon(color='white', icon_color=row['marker_color']))
    marker_cluster.add_child(launch_marker)
site_map
```

- Added mouse position function map to locate coordinates of launch sites and other points of interest

```
# Add Mouse Position to get the coordinate (Lat, Long) for a mouse over on the map
formatter = "function(num) {return L.Util.formatNum(num, 5);};"
mouse_position = MousePosition(
    position='topright',
    separator=' Long: ',
    empty_string='NaN',
    lng_first=False,
    num_digits=20,
    prefix='Lat:',
    lat_formatter=formatter,
    lng_formatter=formatter,)
site_map.add_child(mouse_position)
site_map
```

- Created a polyline to display distances between launch sites and the nearest coastline

```
distance_line=folium.PolyLine(
    locations=[launch_site_coordinates, coastline_coordinates],
    weight=1)
site_map.add_child(distance_line)
site_map
```

Interactive Dashboard with Plotly

Github Interactive Dashboard with Plotly Link

Generated interactive scatter plots and pie charts displaying rocket payload ranges and successful launches

- Created the layout for the application

```
app.layout = html.Div(children=[html.H1('SpaceX Launch Records Dashboard',
                                        style={'textAlign': 'center', 'color': '#503D36',
                                               'font-size': 40}),
```

- Added a pie chart, with drop down menus, displaying successful launch information

```
html.Div(dcc.Graph(id='success-pie-chart')),
html.Br(),

html.P("Payload range (Kg):"),
```

- Assembled an interactive scatter plot chart with options to review various launch sites with corresponding payload weights

```
@app.callback( Output(component_id='success-payload-scatter-chart', component_property='figure'),
              [Input(component_id='site-dropdown', component_property='value'),
               Input(component_id='payload-slider',component_property='value')])
def get_payload_chart(launch_site, payload_mass):
    if launch_site == 'All Sites':
        fig = px.scatter(spacex_df[spacex_df['Payload Mass (kg)'].between(payload_mass[0], payload_mass[1])],
                x="Payload Mass (kg)",
                y="class",
                color="Booster Version Category",
                hover_data=['Launch Site'],
                title='Correlation Between Payload and Success for All Sites')
    else:
        df = spacex_df[spacex_df['Launch Site']==str(launch_site)]
        fig = px.scatter(df[df['Payload Mass (kg)'].between(payload_mass[0], payload_mass[1])],
                x="Payload Mass (kg)",
                y="class",
                color="Booster Version Category",
                hover_data=['Launch Site'],
                title='Correlation Between Payload and Success for Site {}'.format(launch_site))
    return(fig)
```

Predictive Analytics with Machine Learning

[Github Predictive Analytics with Machine Learning Link]

Assembled and tested various classification models to determine which algorithm will be the most accurate for further launch predictions

- Created training and testing perimeters based on collected EDA information

```
Y = data['Class'].to_numpy()
Y
```

```
X = transform.fit_transform(X)
```

```
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.2, random_state=2)
print ('Train set:', X_train.shape,  Y_train.shape)
print ('Test set:', X_test.shape,  Y_test.shape)
```

- Determined which launches were successful with a payload mass between 4000 and 600 kg

```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
lr=LogisticRegression()
```

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()
logreg_cv = GridSearchCV(lr,parameters,cv=10)
logreg_cv.fit(X_train, Y_train)
```

- Created a confusion matrix visual based on the test set accuracy percentage

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

Predictive Analytics with Machine Learning - Continued

- Put together a decision tree classifier object with the training and testing data

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

```
gridsearch_cv_tree = GridSearchCV(tree, parameters, scoring='accuracy', cv=10)
tree_cv = gridsearch_cv_tree.fit(X_train, Y_train)
```

- Added a K Nearest Neighbors object classifier for additional testing

```
parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'p': [1,2]}

KNN = KNeighborsClassifier()
```

```
knn_cv = GridSearchCV(KNN,parameters,cv=10)
knn_cv.fit(X_train, Y_train)
```

- Calculated the accuracy of the K Nearest Neighbors tree classifier

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

Summary of Results

Exploratory Data Analysis

- Designated Orbit Types for Falcon 9 Launches
- Falcon 9 Landing Outcome Definitions and Launch Results
- SQL Launch Analysis Results

Visual Analytics

- EDA Launch Results
- Interactive Launch Site Maps
- Interactive Visual Dashboard

Predictive Analysis

- Classification Accuracy
- Confusion Matrix

Results

Exploratory Data Analysis

Designated Orbit Types for Falcon 9 Launches

LEO : Low Earth orbit (LEO)is an Earth-centred orbit with an altitude of 2,000 km (1,200 mi) or less (approximately one-third of the radius of Earth),[1] or with at least 11.25 periods per day (an orbital period of 128 minutes or less) and an eccentricity less than 0.25.[2] Most of the manmade objects in outer space are in LEO.

VLEO : Very Low Earth Orbits (VLEO) can be defined as the orbits with a mean altitude below 450 km. Operating in these orbits can provide a number of benefits to Earth observation spacecraft as the spacecraft operates closer to the observation.

GTO : A geosynchronous orbit is a high Earth orbit that allows satellites to match Earth's rotation. Located at 22,236 miles (35,786 kilometers) above Earth's equator, this position is a valuable spot for monitoring weather, communications and surveillance. Because the satellite orbits at the same speed that the Earth is turning, the satellite seems to stay in place over a single longitude, though it may drift north to south," NASA wrote on its Earth Observatory website.

SSO (or SO) : It is a Sun-synchronous orbit also called a helio-synchronous orbit is a nearly polar orbit around a planet, in which the satellite passes over any given point of the planet's surface at the same local mean solar time.

ES-L1 :At the Lagrange points the gravitational forces of the two large bodies cancel out in such a way that a small object placed in orbit there is in equilibrium relative to the center of mass of the large bodies. L1 is one such point between the sun and the earth.

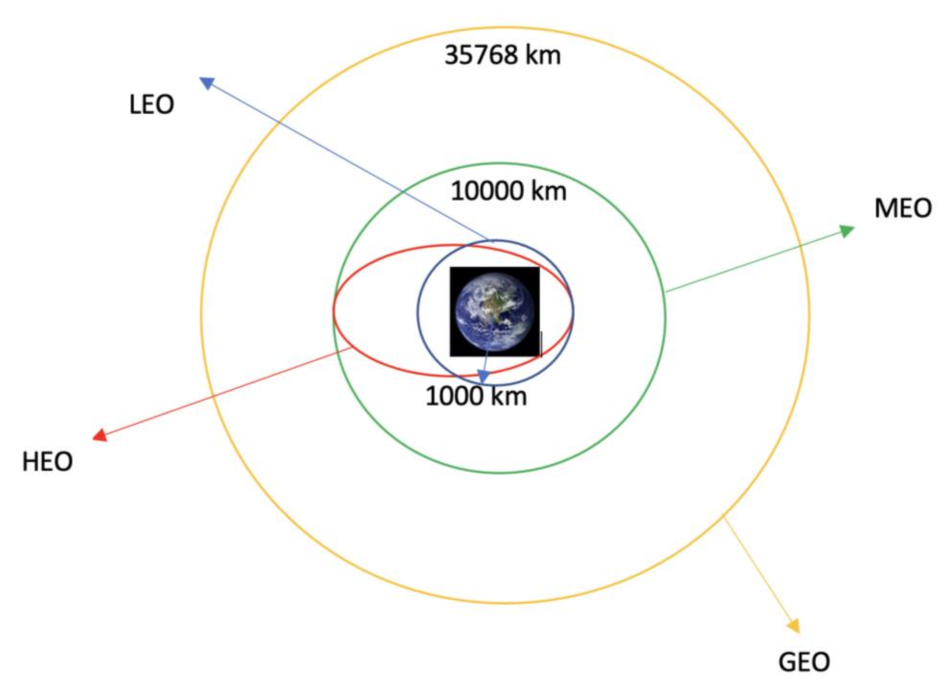
HEO A highly elliptical orbit, is an elliptic orbit with high eccentricity, usually referring to one around Earth.

ISS A modular space station (habitable artificial satellite) in low Earth orbit. It is a multinational collaborative project between five participating space agencies: NASA (United States), Roscosmos (Russia), JAXA (Japan), ESA (Europe), and CSA (Canada)

MEO Geocentric orbits ranging in altitude from 2,000 km (1,200 mi) to just below geosynchronous orbit at 35,786 kilometers (22,236 mi). Also known as an intermediate circular orbit. These are "most commonly at 20,200 kilometers (12,600 mi), or 20,650 kilometers (12,830 mi), with an orbital period of 12 hours

HEO Geocentric orbits above the altitude of geosynchronous orbit (35,786 km or 22,236 mi)

GEO It is a circular geosynchronous orbit 35,786 kilometers (22,236 miles) above Earth's equator and following the direction of Earth's rotation



Exploratory Data Analysis

Falcon 9 Landing Outcome Definitions and Launch Results

True Ocean means the mission outcome was successfully landed to a specific region of the ocean.

False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean.

True RTLS means the mission outcome was successfully landed to a ground pad.

False RTLS means the mission outcome was unsuccessfully landed to a ground pad.

True ASDS means the mission outcome was successfully landed to a drone ship.

False ASDS means the mission outcome was unsuccessfully landed to a drone ship.

None ASDS and **None None** represent a failure to land.

Total Launches by Orbit Type :

GTO	27
ISS	21
VLEO	14
PO	9
LEO	7
SSO	5
MEO	3
ES-L1	1
HEO	1
SO	1
GEO	1

Total Launches by Landing Outcome :

True ASDS	41
None None	19
True RTLS	14
False ASDS	6
True Ocean	5
False Ocean	2
None ASDS	2
False RTLS	1

Exploratory Data Analysis

SQL Launch Analysis Results

Total Payload Mass Carried by Boosters Launched by NASA :

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass Carried by Falcon F9 v1.1 :

AVG(PAYLOAD_MASS__KG_)
2928.4

Total Number of Successful and Failed Mission Launches :

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

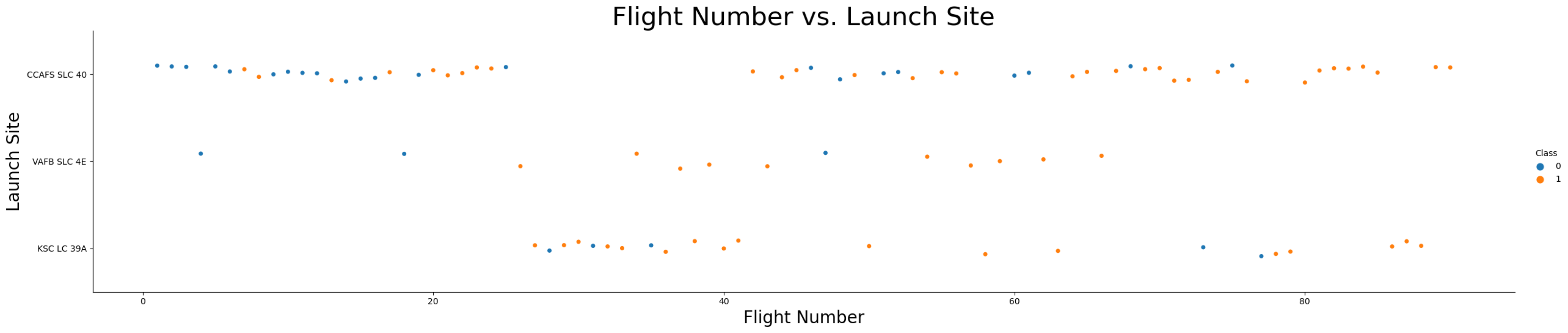
Landing Outcome by Rank - 2010 to 2017 :

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

Visual Analytics

EDA Launch Results

Visualization of the Relationship between Flight Number and Launch Site:

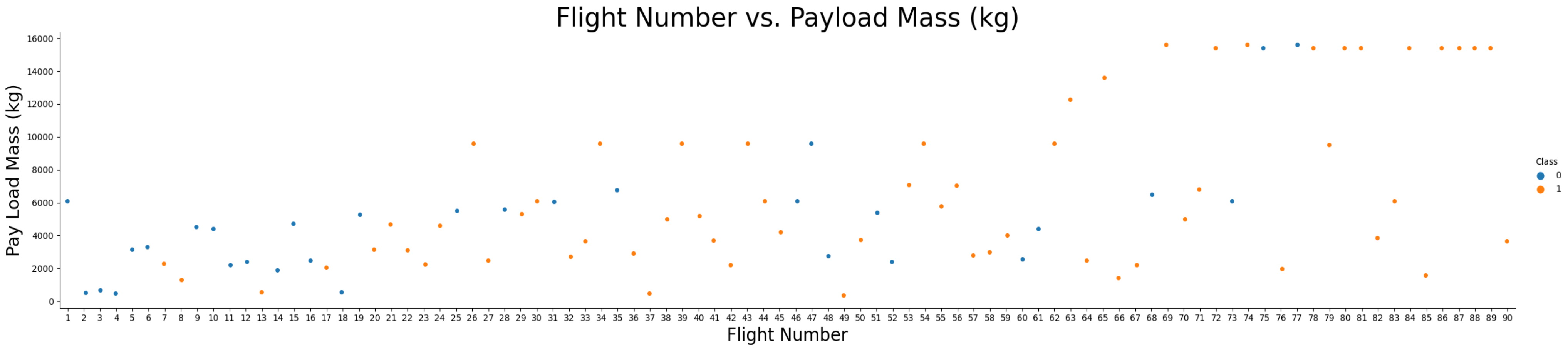


- As the number of flights increased the success rate at each launch site also improved.
- The majority of early launches originated from the CCAFS SLC 40 site with a low success rate.
- Following Flight Number 30 the number of successful launches began to noticeably improve.

Visual Analytics

EDA Launch Results Continued

Visualization of the Relationship between Flight Number and Payload Mass :

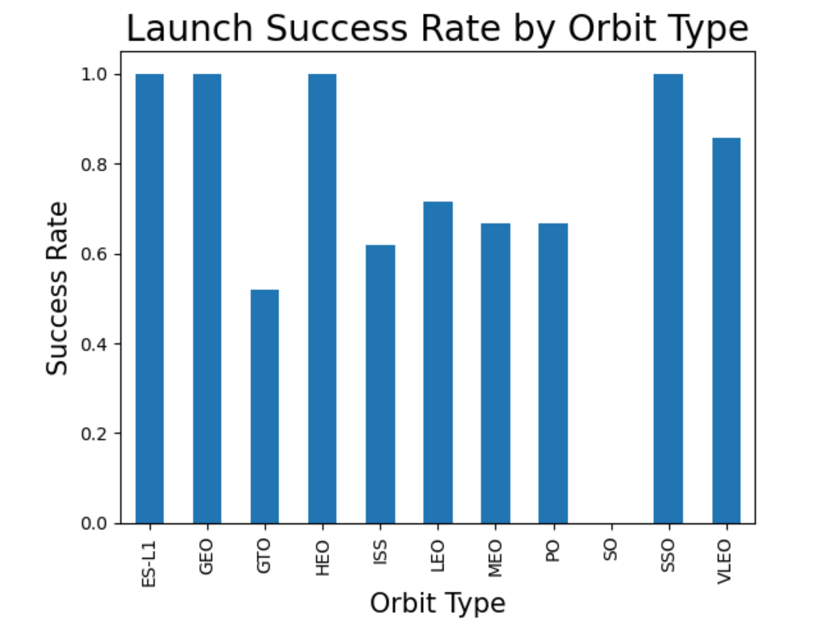


- On average, the larger the payload mass, the higher the success rate of the launch became.
- However, most of the launches that had a payload greater than 7,000 kg were not successful.
- The KSC LC 39A launch site had a 100% success rate for payloads less than 5,500 kg.

Visual Analytics

EDA Launch Results Continued

Visualization of the Launch Success Rate Based on Destination Orbit :

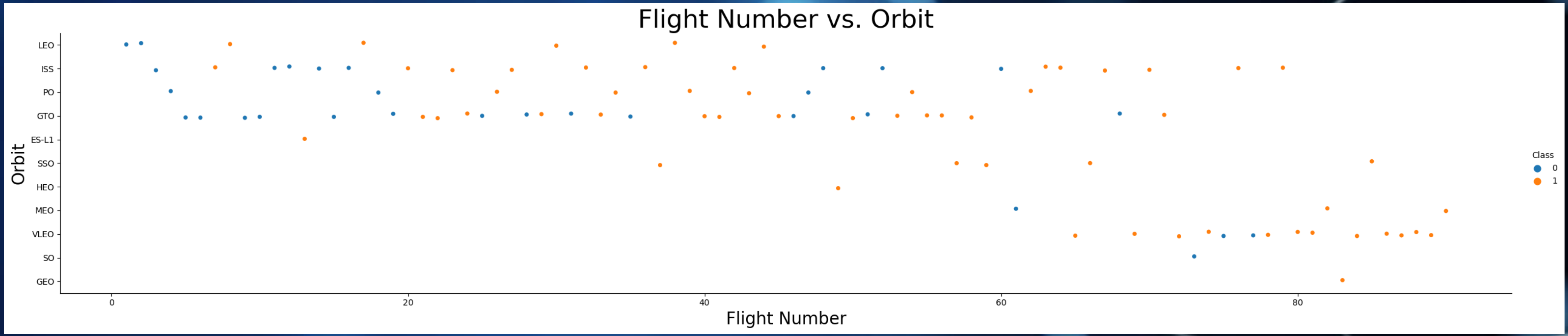


- Sun-synchronous (SO), or helio-synchronous, orbit launches encountered a 0% success rate.
- GTO, ISS, LEO, MEO, PO orbits had a moderate success rate (50 – 80%).
- All ES L1, GEO, HEO and SSO orbits were successful with all launch attempts (100%).

Visual Analytics

EDA Launch Results Continued

Visualization of the Relationship between Flight Number and Orbit Destination :

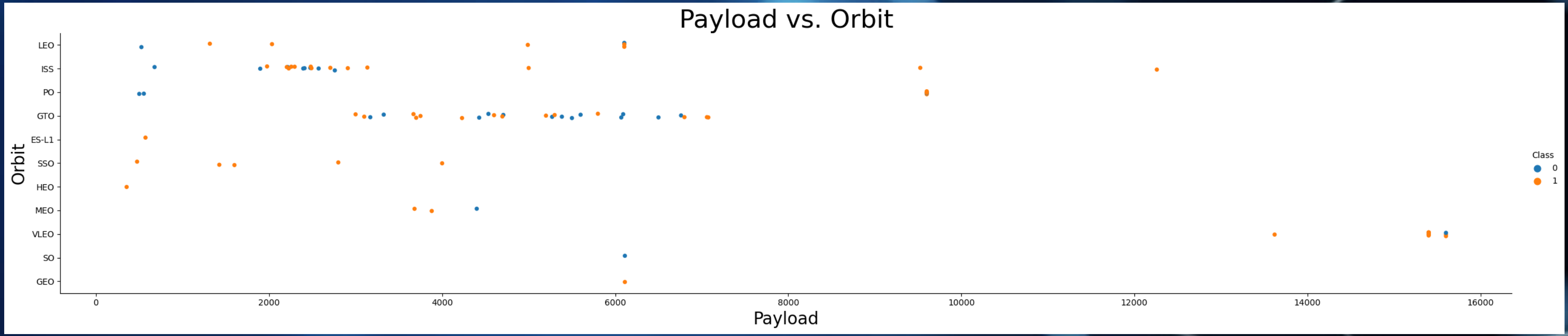


- Overall flight success improved as the number of launches to each orbit type increased.
- The success of the GEO, HEO, and ES L1 flights can be attributed to there being only one launch per each orbit type.
- Flights to SSO was the most effective of the orbit types with five successful launches.

Visual Analytics

EDA Launch Results Continued

Visualization of the Relationship between Payload Mass and Orbit Destination :

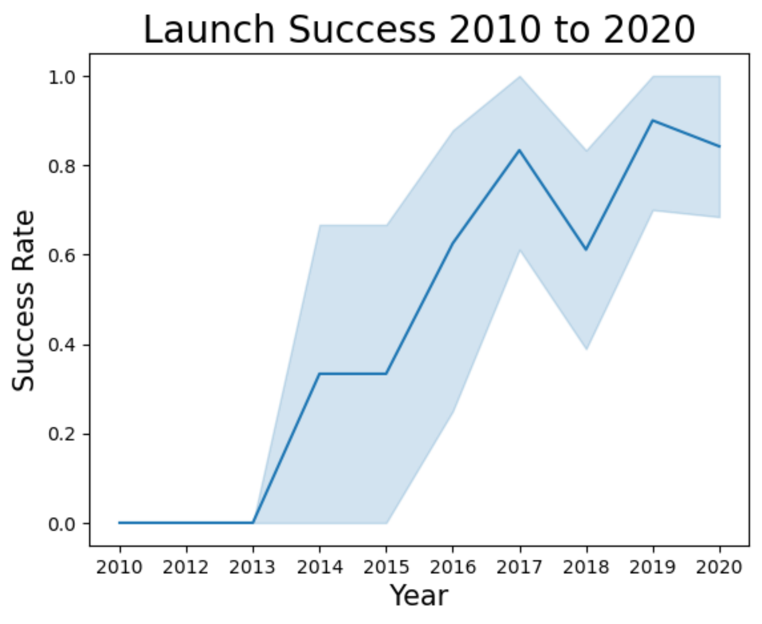


- LEO, ISS, and PO orbit destinations have had better success carrying larger payloads.
- GTO orbit flights have had close to a fifty percent success rate with each payload mass.
- VLEO had the highest success rate with the largest payload flights.

Visual Analytics

EDA Launch Results Continued

Falcon 9 Launch Success Timeline :

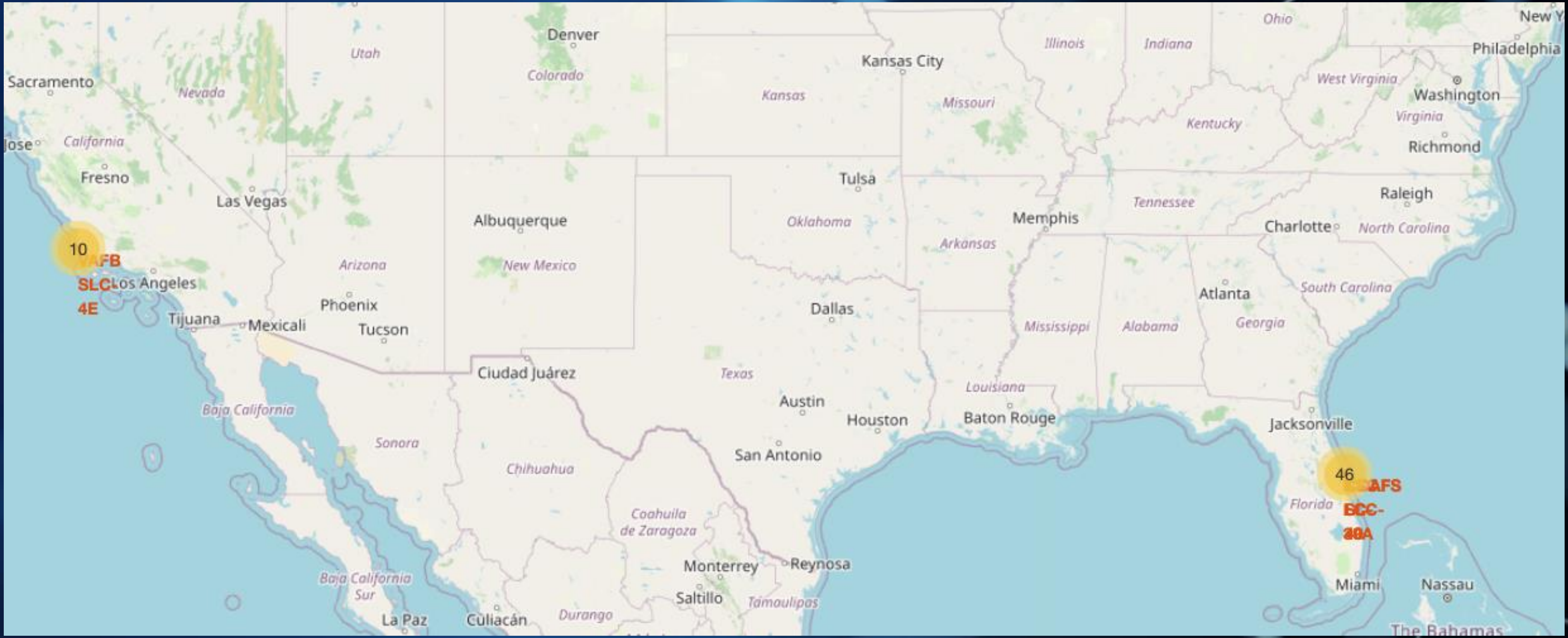


- Between 2010 and 2013 there were no successful Falcon 9 launches.
- After 2013, the success outcome of launches greatly increased.
- Success rates decreased between 2017 and 2018 and trended upward in 2019.

Visual Analytics

Folium Interactive Launch Site Map

SpaceX Launch Site Locations :

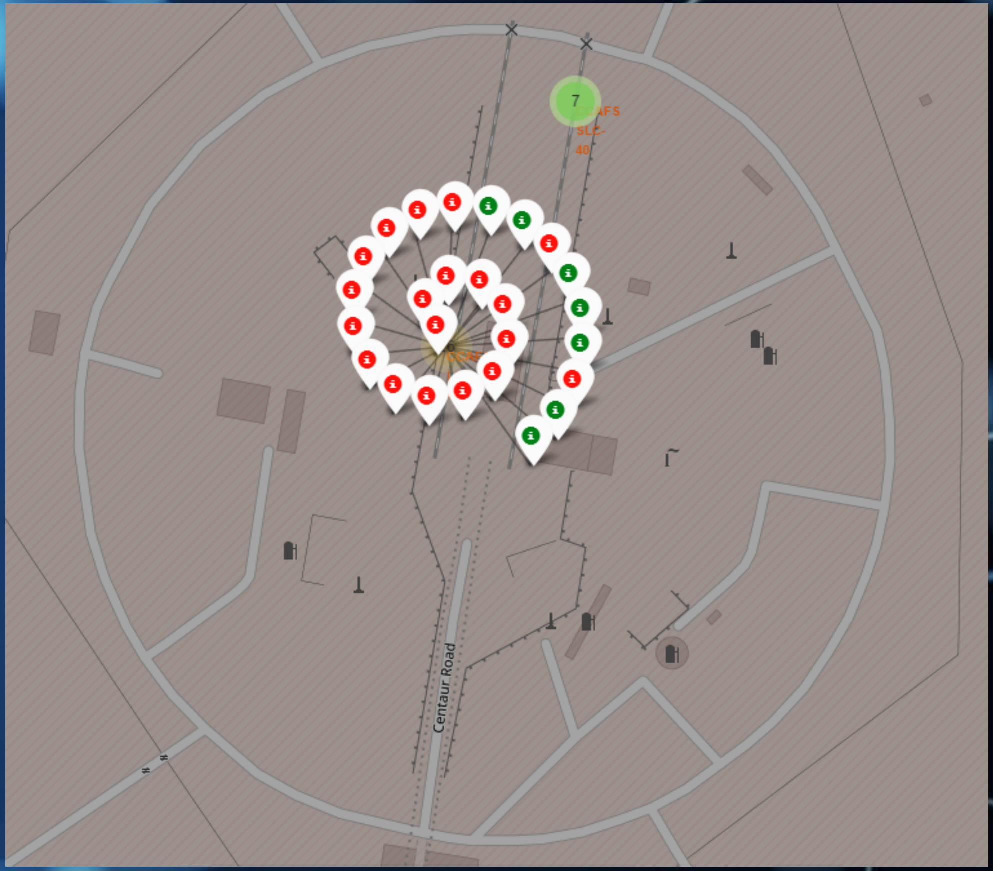


IBM Data Science Capstone Project 2023 - Charles Schultz

Visual Analytics

Folium Interactive Launch Site Map Continued

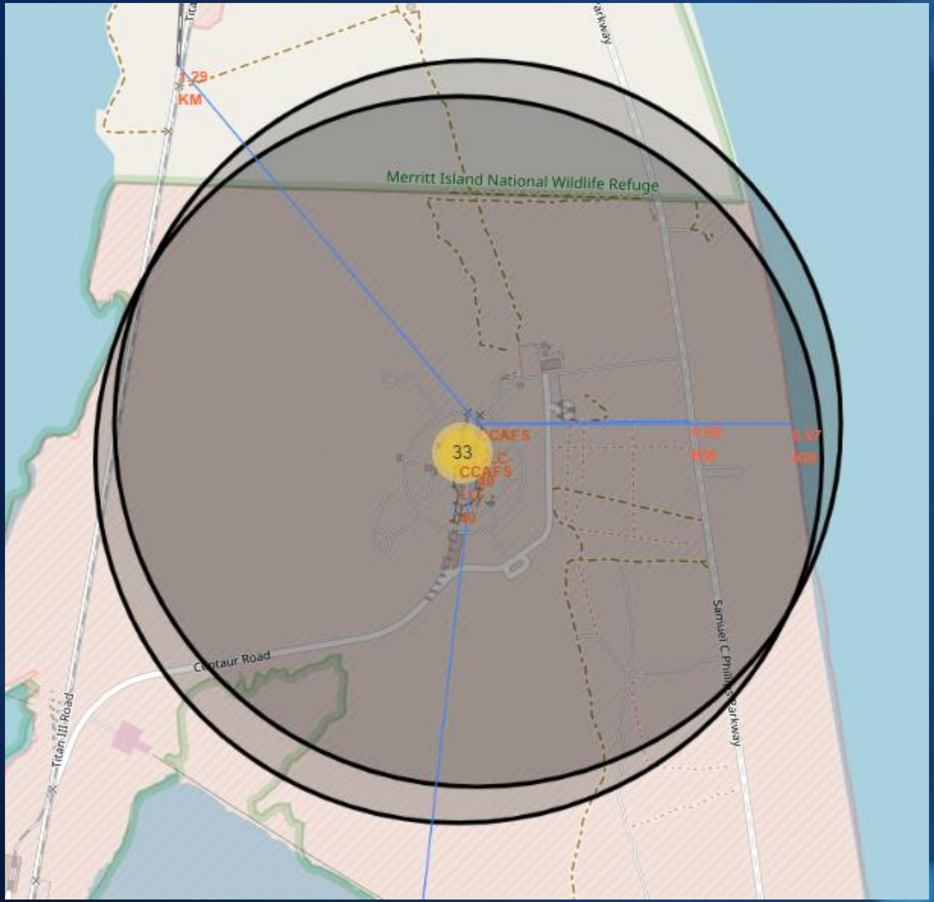
Launches by Location with Corresponding Launch Outcomes :



Visual Analytics

Folium Interactive Launch Site Map Continued

Launch Sites in Proximity of Railways, Highways, and Coastline:

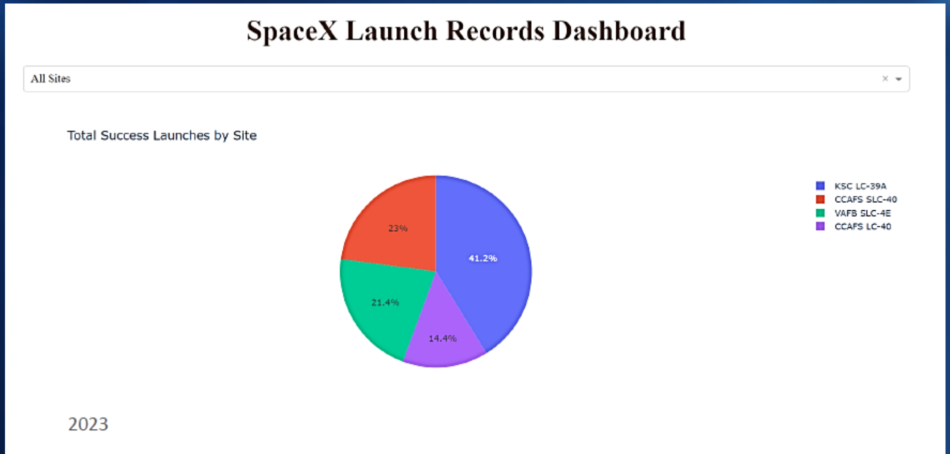


- SpaceX launch sites are primarily located on California’s Pacific Coast and Florida’s Atlantic Coast.
- Launch sites are positioned as close as possible to the equator within the borders of the United States.
- Sites are in close proximity to railways, highways, and coastlines but constructed safely away from highly populated areas.

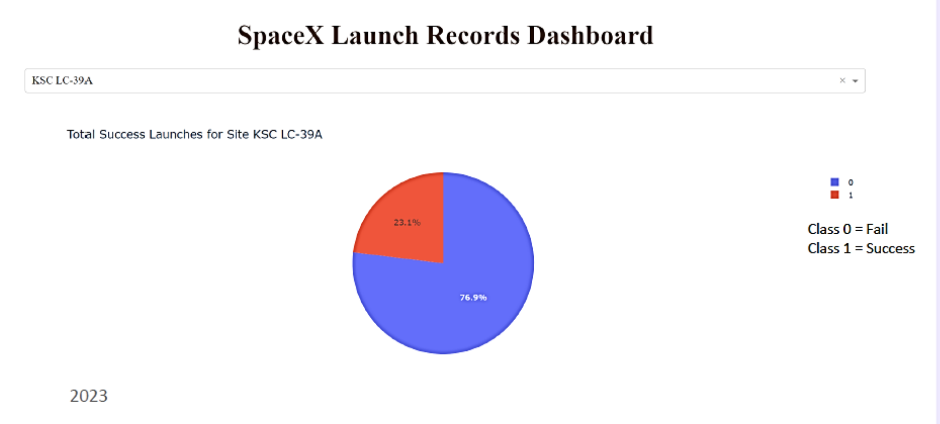
Visual Analytics

Plotly Interactive Dashboard

Successful Flights Based on Launch Site :



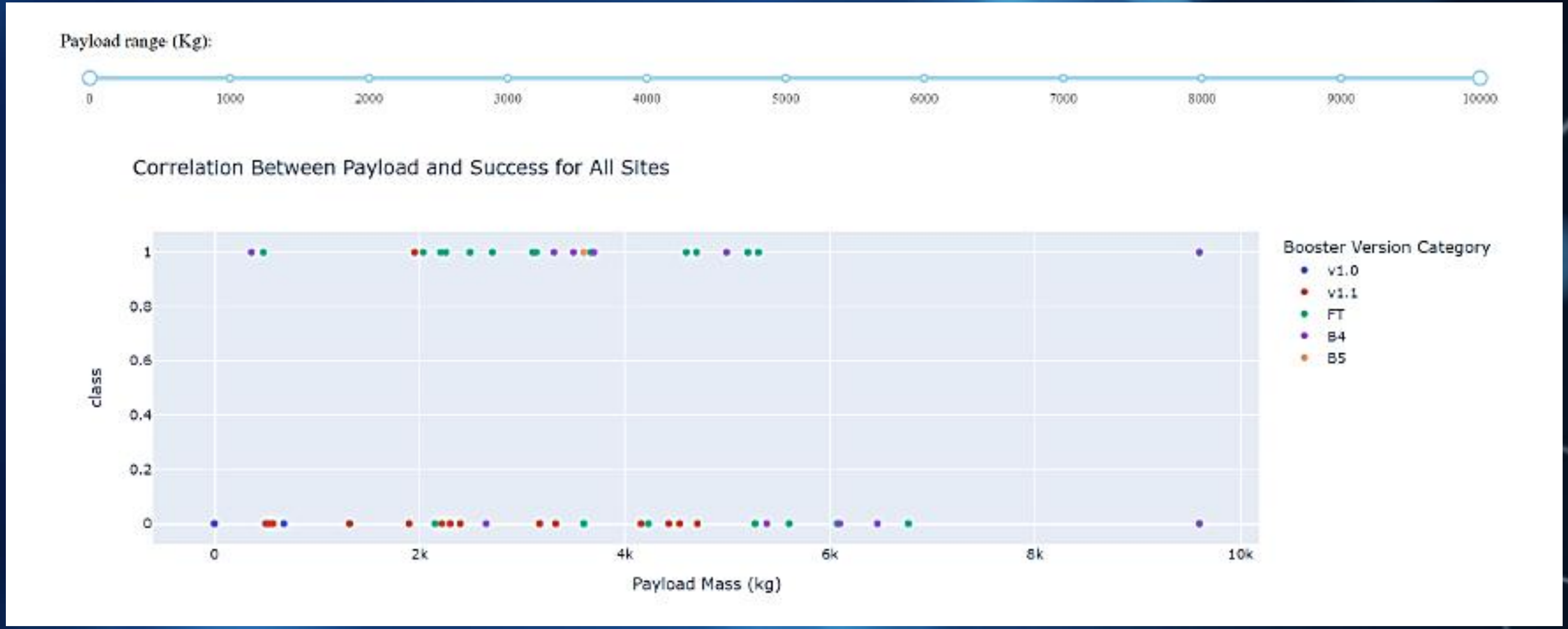
Successful Flights at KSC LC-39A Site :



Visual Analytics

Plotly Interactive Dashboard Continued

Relationship Between Payload Mass and Launch Site Locations :



- KSC LC 39 A was the most successful of the launch sites with a 41.2% success rating.
- KSC LC 39 A also had the highest rate of successful launches from all the sites with a 76.9% success rate.
- Flights carrying a payload mass between 2,000 and 5,000 kg were the most successful of the launches.

Predictive Analysis

Classification Accuracy

Logistic Regression Classification Model :

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

Support Vector Classifier Classification Model :

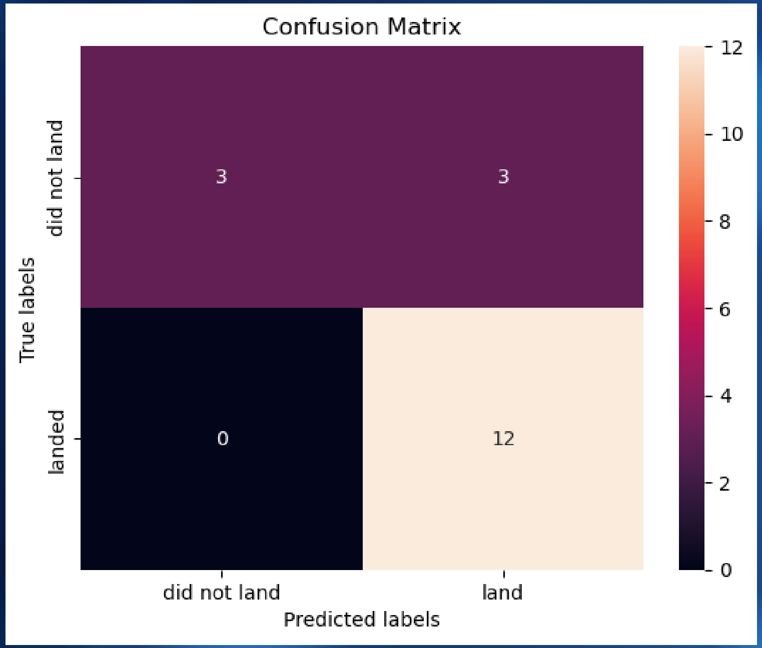
```
GridSearchCV
GridSearchCV(cv=10, estimator=SVC(),
             param_grid={'C': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
       1.00000000e+03]),
                         'gamma': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
       1.00000000e+03]),
                         'kernel': ('linear', 'rbf', 'poly', 'rbf', 'sigmoid')})
estimator: SVC
SVC
```

K Nearest Neighbor Classification Model :

```
GridSearchCV
GridSearchCV(cv=10, estimator=KNeighborsClassifier(),
             param_grid={'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                         'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
                         'p': [1, 2]})
estimator: KNeighborsClassifier
KNeighborsClassifier
```

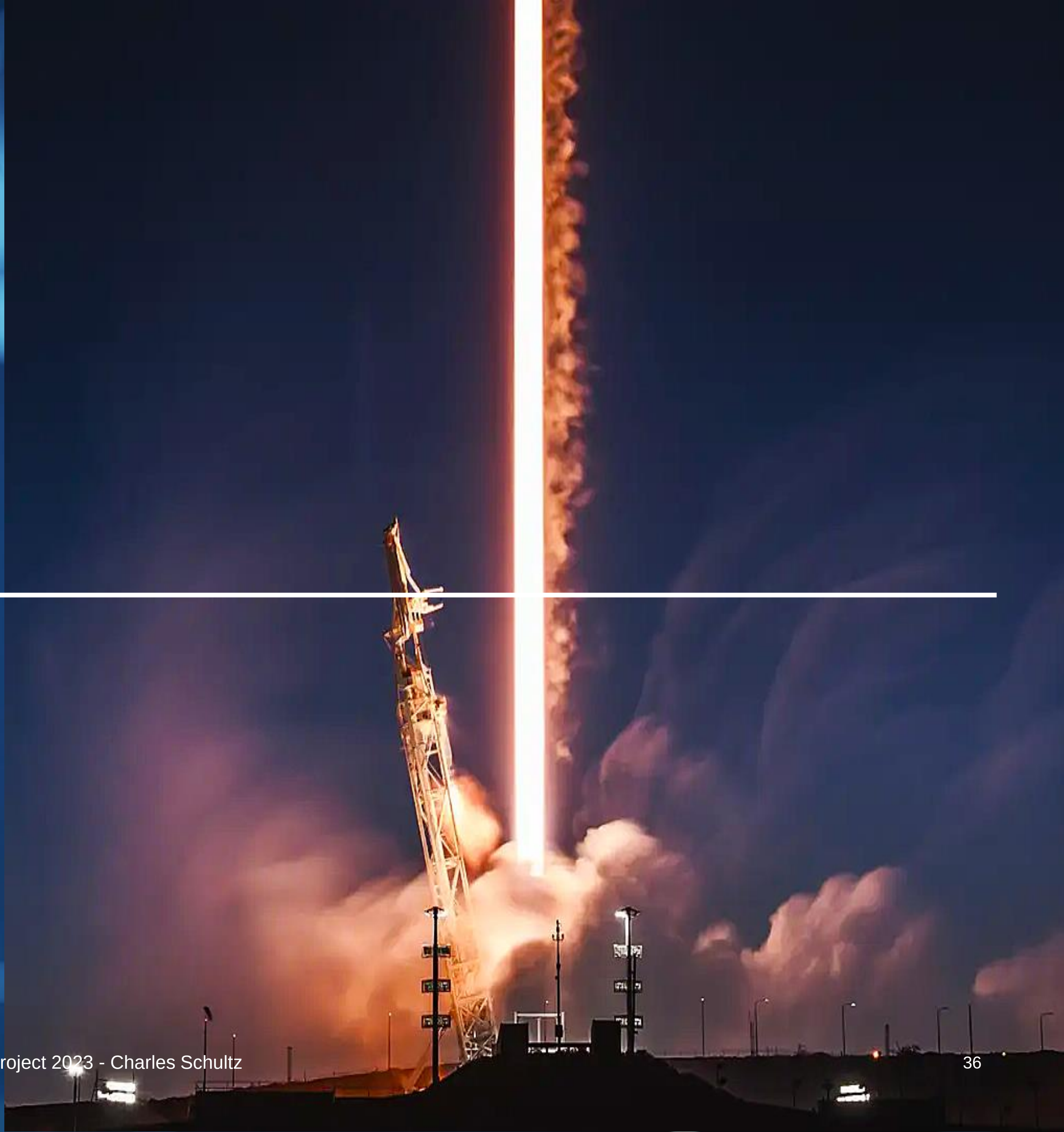
Predictive Analysis
Confusion Matrix

Relationship Between Payload Mass and Launch Site Locations :



- Logistic Regression, SVM, Decision Tree, and K Nearest Neighbor datasets were trained and tested with closely similar results for each model.
- The Decision Tree model proved to be the best option with a 94.44% accuracy rating.
- Tested Confusion Matrices had identical results. (12 True Positive, 3 True Negative, 3 False Positive, and Zero False Negative)

Conclusion



SpaceX Falcon 9 Rocket Analysis - Conclusion

Evolution of Success :

Our Falcon 9 flight analysis demonstrates a clear trend between experience and success. As the number of flights increased, the margin of error diminished considerably. Initial flight tests between 2010 and 2013 encountered a 0% landing success rate, while the subsequent years established an upward trend of effective launches, albeit marked by minor setbacks in 2018 and 2020.

Orbit Patterns :

Tested flight paths for the Falcon 9 rocket – ES-L1, GEO, HEO, and SSO – demonstrated a 100% success rate. However, the success of this trend can be attributed to the limited number of flights into each of these orbits. Notably, SSO (sun-synchronous orbit) flights stand out with an impressive 100% success rate across five launches.

Site and Success :

Launchpad KSC LC-39A emerged with the highest rate of successful launches (76.9%), along with the overall highest success rate in comparison to other launch sites (41.2%). Visual analytics reveal that the positioning of launch sites near the equator capitalized on Earth's rotational speed, resulting in reduced fuel and cost burdens.

Payload Matters :

Launches destined for VLEO (Very Low Earth Orbit) demonstrate a correlation between heavier payloads and increased flight success. However, GTO (geosynchronous orbit) launches scored close to a fifty percent success rate regardless of the payload mass. These relationships provide opportunities for optimizing payload design and distribution.

Predictive Modeling :

Logistic Regression, SVM, Decision Tree, and K Nearest Neighbor predictive models reveal an intriguing alignment in their results. Nevertheless, the Decision Tree model emerges as the superior performer, showcasing an exceptional accuracy rating of 94.44%. This outcome demonstrates a strong foundation for harnessing predictive power in future missions.

Thank You