Predicting SpaceX Launch Outcomes with Machine Learning

IBM Applied Data Science Capstone

Lucy Herr

February 24, 2022

Executive Summary

Many factors can influence the potential success of a SpaceX Falcon 9 rocket launch. Understanding the most important features of launches and using this information to more accurately predict successful landings is essential to understanding SpaceX's competitive advantage.

This (IBM) capstone project presentation uses SpaceX launch data to identify key features of Falcon 9 launches and build machine learning classification models to predict launch outcomes.

The data used in this project was collected from the SpaceX API and historical Falcon 9 launch record tables on Wikipedia. After data cleaning and wrangling, exploratory data analysis (EDA) and interactive visualizations were used to examine key patterns in the launch data, such the relationships between rocket payload mass, booster version, flight number, launch orbit, and launch site location.

The goal of the predictive analysis was to find the most accurate classification method for predicting launch outcomes. The dataset was used to train and test four models: Logistic Regression, SVM, Decision Tree, and KNN. The Decision Tree model featured the highest accuracy score in predicting Falcon 9 launches; additional insights and implications are also discussed.

Table of Contents

Introduction

This capstone project presentation uses SpaceX launch data to identify key features of Falcon 9 launches and use machine learning classification models to predict successful landing outcomes.

Business Context & Problem:

While rocket launches from other providers can cost upwards of $165 million, SpaceX advertises Falcon 9 rocket launches costing only $62 million. The substantially lower cost of SpaceX launches relates to the company’s innovative approach to reusing the first stage of the rocket. Accurately predicting whether Falcon 9 landings will succeed is critical to SpaceX's ability to launch rockets at lower costs.

Many factors, including flight number, payload mass, orbit type, and launch site location, can influence the potential success of a Falcon 9 launch. Understanding the most important factors of successful launches and using this information to more accurately predict successful landings will enable SpaceX to grow its competitive advantage. Conversely, this information would also be highly useful for alternative launch providers seeking to bid against SpaceX.

Business Questions

- What key features of the Falcon 9 launches can enable us to understand why launches succeed or fail? How do these variables relate to one another?
- Based on this data, which ML classification model can most accurately predict SpaceX launch outcomes?

METHODOLOGY

Methodology: Data Collection

The goal of data collection in this project was to gather the Falcon 9 launch data needed to answer the project's key questions about launch characteristics and outcomes.

DATA SOURCE 1: SpaceX launch data (SpaceX REST API)

Step 1: gather data on rocket type, payload, launch & landing specifications, and landing outcome

- perform get request using the requests library to obtain launch data

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

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

- call .json() method to retrieve a list of JSON objects for each launch
- convert JSON to dataframe (flat table) using json_normalize function

Step 2: use the API to append additional launch information to the dataframe

- use Launch ID to add booster, payload mass, orbit, site longitude and latitude, landing type, etc.
  - Example: gather Launch Pad longitude, latitude, and launch site data

```
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```

- subset dataframe to include only relevant features, rockets, core & payload data, and record dates

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

List of Falcon 9 and Falcon Heavy launches

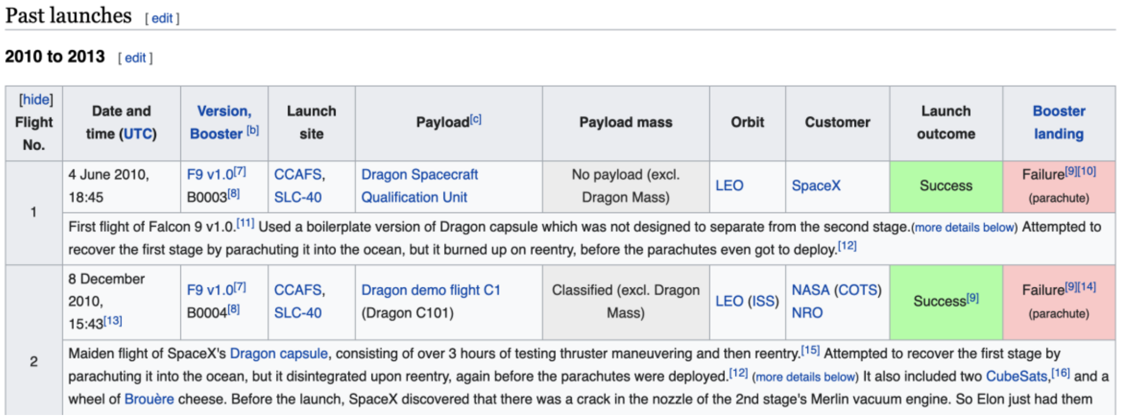
From Wikipedia, the free encyclopedia

Since June 2010, rockets from the Falcon 9 family have been launched 146 times, with 144 full mission successes, one partial failure and one total loss of the spacecraft. In addition, one rocket and its payload were destroyed on the launch pad during the fueling process before a static fire test was set to occur.

Designed and operated by private manufacturer SpaceX, the Falcon 9 rocket family includes the retired versions Falcon 9 v1.0, v1.1, and v1.2 "Full Thrust" Block 1 to 4, along with the currently active Block 5 evolution. Falcon Heavy is a heavy-lift derivative of Falcon 9, combining a strengthened central core with two Falcon 9 first stages as the side boosters.[1]

The Falcon design features reusable first-stage boosters, which land either on a ground pad near the launch site or on a drone ship at sea.[2] In December 2015, Falcon 9 became the first rocket to land propulsively after delivering a payload into orbit.[3] This achievement significantly reduced launch costs.[4] Falcon family core boosters have successfully landed 109 times in 120 attempts. A total of 29 boosters have flown multiple missions, with a record of eleven missions by the same booster.

Left to right: Falcon 9 v1.0, v1.1, v1.2 "Full Thrust", Falcon 9 Block 5, Falcon Heavy, and Falcon Heavy Block 5.



Past launches [edit]

2010 to 2013 [edit]

[hide] Flight No.	Date and time (UTC)	Version, Booster [b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
1	4 June 2010, 18:45	F9 v1.0[7] B0003[8]	CCAFS, SLC-40	Dragon Spacecraft Qualification Unit	No payload (excl. Dragon Mass)	LEO	SpaceX	Success	Failure[9][10] (parachute)
	First flight of Falcon 9 v1.0.[11] Used a boilerplate version of Dragon capsule which was not designed to separate from the second stage.(more details below) Attempted to recover the first stage by parachuting it into the ocean, but it burned up on reentry, before the parachutes even got to deploy.[12]								
2	8 December 2010, 15:43[13]	F9 v1.0[7] B0004[8]	CCAFS, SLC-40	Dragon demo flight C1 (Dragon C101)	Classified (excl. Dragon Mass)	LEO (ISS)	NASA (COTS) NRO	Success[9]	Failure[9][14] (parachute)
	Maiden flight of SpaceX's Dragon capsule, consisting of over 3 hours of testing thruster maneuvering and then reentry.[15] Attempted to recover the first stage by parachuting it into the ocean, but it disintegrated upon reentry, again before the parachutes were deployed.[12] (more details below) It also included two CubeSats,[16] and a wheel of Brouère cheese. Before the launch, SpaceX discovered that there was a crack in the nozzle of the 2nd stage's Merlin vacuum engine. So Elon just had them								

DATA SOURCE 2: historical Falcon 9 launch records (Wikipedia)

- Use Python BeautifulSoup package to webscrape HTML tables of past Falcon 9 launch records
  - Source: Wikipedia page, "List of Falcon 9 and Falcon Heavy launches"
  - Use HTTP GET method to request the Falcon9 Launch HTML page as an HTTP response
- Create a BeautifulSoup object from the HTML response
  - Extract variable names from the HTML table headers
- Parse data from tables
  - Create an empty dictionary with keys from the extracted column names
  - Fill launch dictionary with launch records extracted from table rows
  - Convert into Pandas dataframe for visualization and predictive analysis

Methodology: Data Wrangling

The goal of data wrangling in this project was to prepare and format the SpaceX data for EDA and predictive analysis.

- Filter data to include Falcon 9 data only (exclude Falcon 1 launches)
- Address null values:
  - Payload Mass: impute null values with mean payload mass
  - Landing Pad: retain nulls (when landing pad is not used) until one-hot encoding
- Convert landing outcomes into Training Labels
  - Of the 8 landing outcome types, 4 represent failed outcomes

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

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

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

  - Convert into "Landing Class" labels: 0=failed landing, 1=successful landing

```
# create list in which element=0 if bad_outcome & element = 1 if good outcome
landing_class = []
for i in df['Outcome']:
    if i in set(bad_outcomes):
        landing_class.append(0)
    else:
        landing_class.append(1)
```

Methodology: EDA & Interactive Visual Analytics

The goal of EDA and data visualization in this project was to understand important relationships between the many factors related to Falcon 9 launch outcomes and to identify key variables to use in predictive analysis.

- **EDA - SQL:** use SQL to develop preliminary understanding of the SpaceX data
  - Establish connection to SpaceX dataset in DB2 & use SQL within Jupyter notebook
  - Queries included listing the unique launch sites in the data and ranking totals of landing outcomes types
- **EDA - Visualization:** explore visual trends in the data to evaluate relevant variables for modeling
  - Use matplotlib to generate scatter plots, bar charts, and time series charts
  - Examine relationships between variables including payload mass, orbit type, flight number, and launch site
- **EDA - Launch Site Location Analysis:** examine geographical features of SpaceX launch site locations
  - Create a Folium interactive map to identify geographical characteristics of the four SpaceX launch sites (location, proximity, success rate by site)
  - Identify patterns in launch site proximities such as distance from coastlines and cities

Methodology: Predictive Analysis (ML Classification)

The goal of the predictive analysis step was to use machine learning to build and compare classification models that can predict success rates of Falcon 9 launches.

- Feature Engineering: based on preliminary insights from EDA, select variables for prediction
  - Use one-hot encoding for categorical variables (Orbits, LaunchSite, LandingPad, and Serial)
- Standardize the data

```
# Standardize the data in X then reassign it to the variable X
transform = preprocessing.StandardScaler()
```

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

- Split the dataset into training data and test data

```
# split the data X and Y into training and test data.
# Set the parameter test_size to 0.2 and random_state to 2.
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

- Train 4 different classification models: **Logistic Regression**, **SVM**, **Decision Tree**, & **KNN**
  - Hyperparameter grid search and confusion matrix for each model
- Compare the models based on accuracy scores with test data to identify the best-performing method for launch success prediction

RESULTS

Results: EDA with SQL

Launch Site Queries

What are the unique launch sites in the dataset?

```
%sql select DISTINCT(Launch_site) from SPACEXTBL
```

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

- Four unique launch sites will be addressed in the location analysis

What are the first 5 records for launch sites beginning with 'CCA'?

```
%sql SELECT * from SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5;
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Booster Version & Payload Mass Queries

What is the total payload mass carried by boosters launched by NASA (CRS)?

```
%sql SELECT SUM(payload_mass__kg_) AS total_plmkg from SPACEXTBL WHERE customer = 'NASA (CRS)'
```

total_plmkg
45596

What is the average payload mass carried by booster version F9 v1.1?

```
%sql SELECT AVG(payload_mass__kg_) AS avg_plmkg from SPACEXTBL WHERE booster_version LIKE 'F9 v1.0%'
```

avg_plmkg
340

Which boosters have success in drone ship and payload mass between 4,000 and 6,000 kg?

```
%sql select booster_version from SPACEXTBL
WHERE landing__outcome = 'Success (drone ship)'
AND payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000 ;
```

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

Which booster versions have carried the maximum payload mass (15,600 kg)?

```
%sql SELECT DISTINCT(booster_version) from SPACEXTBL
where payload_mass__kg_ =
(SELECT MAX(payload_mass__kg_) as max_plm from SPACEXTBL)
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

- Specific Falcon 9 booster versions may be used based on the launch's landing type (e.g., drone ship) or payload mass

Landing Outcome Queries

What is the total number of mission successes and failures in the dataset?

```
%sql select mission_outcome, COUNT(mission_outcome) AS total_mo from SPACEXTBL GROUP by mission_outcome
```

mission_outcome	total_mo
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- The vast majority of mission outcomes were successful.

When was the first successful landing outcome in ground pad?

```
%sql SELECT MIN(DATE) from SPACEXTBL WHERE landing__outcome = 'Success (ground pad)'
```

1
2015-12-22

In 2015, what were the booster versions and launch sites for failed landings in drone ship?

```
%sql SELECT booster_version, landing__outcome, launch_site from SPACEXTBL
WHERE landing__outcome = 'Failure (drone ship)' AND DATE LIKE '2015%'
```

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

Between 4/6/2010 and 3/20/2017, what were the total landing outcome counts (in descending order)?

```
%sql select landing__outcome, COUNT(landing__outcome) AS count_lo from SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' and '2017-03-20'
GROUP BY landing__outcome ORDER BY count_lo DESC
```

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

- "No attempt" landing outcomes (10) were the most frequent in this time period
- Drone ship landings were also more frequent, with equal failures and successes
- Ground pad landings also had 2 successes

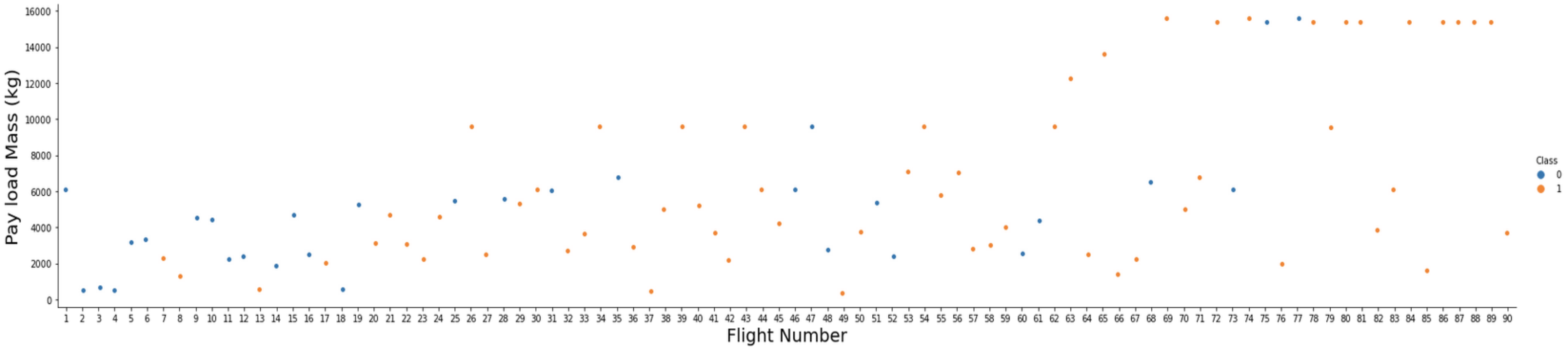
Results:
EDA with Visualization

How do Payload Mass and Flight Number relate to Launch Outcome?

Payload Mass: mass transported by rocket (in kg)

Flight Number: total launch attempts for rocket

Launch Outcome (Class): success = 1, failure = 0



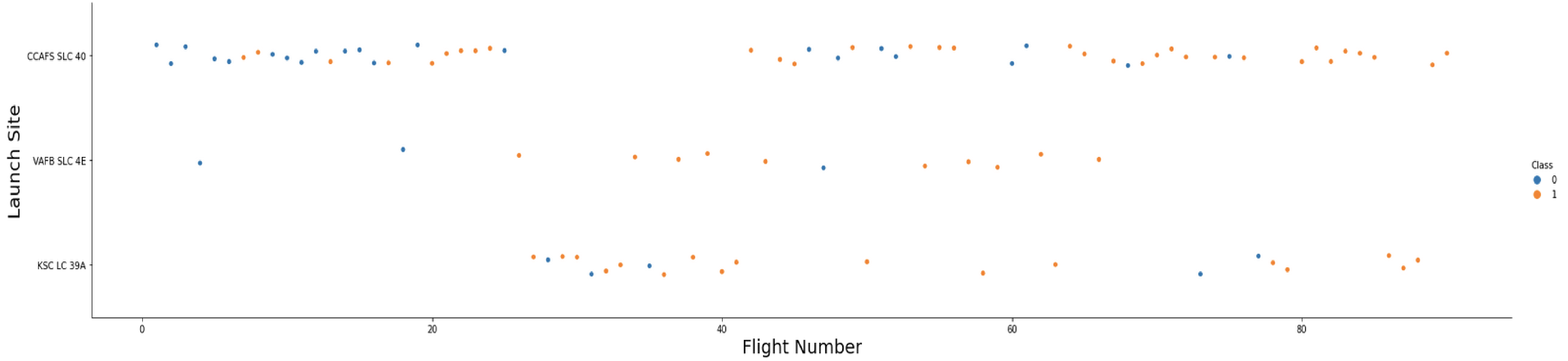
- As flight number increased, the first stage was more likely to land successfully.
- As payload became more massive, the first stage was less likely to land successfully.
- Launch successes varied by site: CCAFS LC-40 had a lower average success rates (60%), while KSC LC-39A and VAFB SLC 4E had higher rates (77%).

How do Launch Site and Flight Number relate to Launch Outcome?

Payload Mass: mass transported by rocket (in kg)

Flight Number: total launch attempts for rocket

Launch Outcome (Class): success = 1, failure = 0

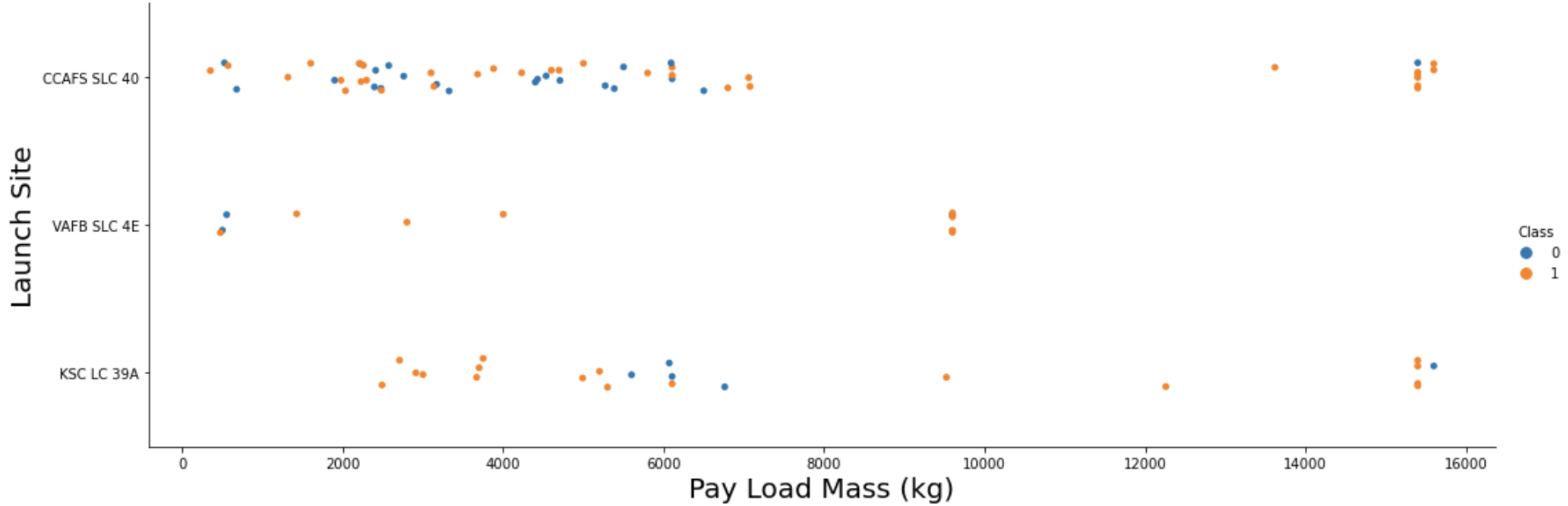


- Typical flight numbers varied by site: at CCAFS SLC-40, launches had lower (0-25) or higher (40-80+) flight numbers. At VAFB SLC-4E, most flight numbers were in the mid-to-high range (25-60); all KSC LC-39A flight numbers were 25 or higher, including a few over 80.
- Launches at CCAFS SLC 40 & VAFV SLC 4E with flight numbers below 20 tended to have more successful outcomes, but a number of CCAFS SLC 40 launches with flight numbers > 40 were also successfully landed. No launches with flight numbers above 80 were successful.

How do Launch Site and Payload Mass relate to Launch Outcome?

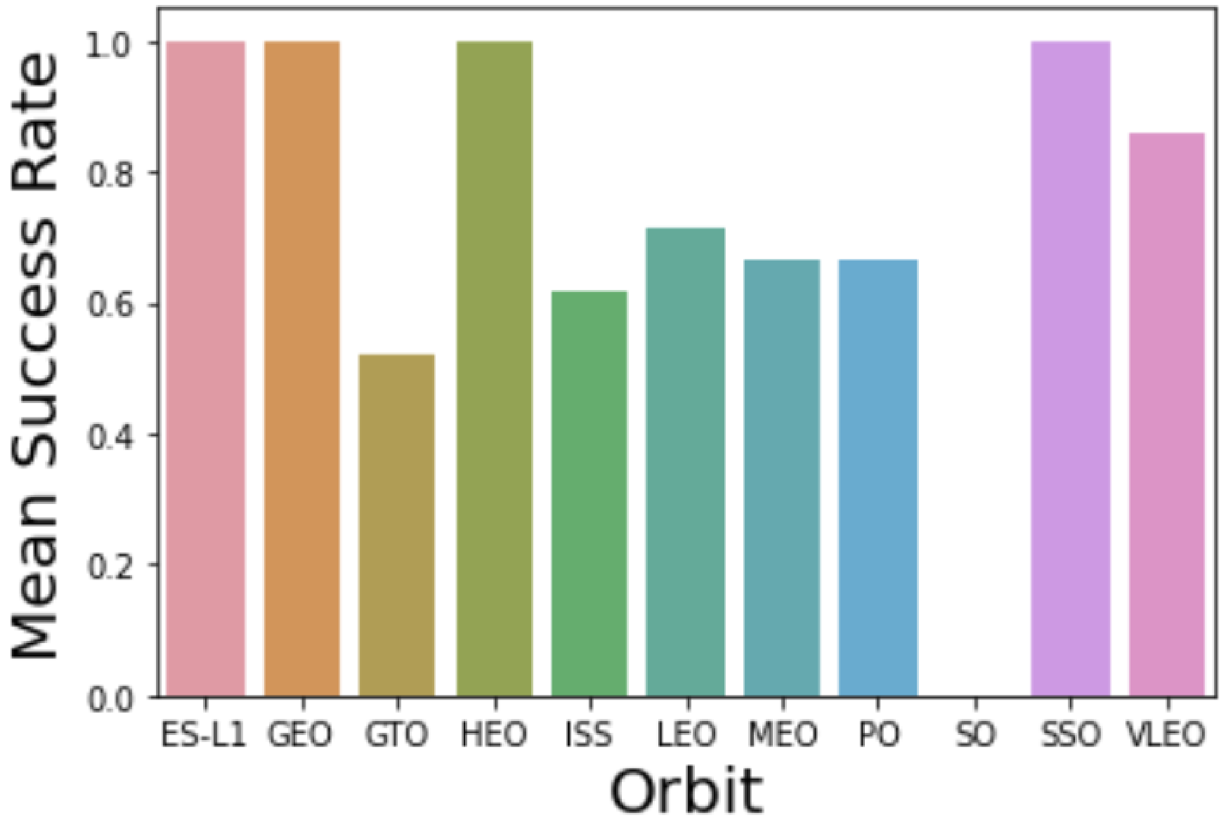
Payload Mass: mass transported by rocket (in kg)

Launch Outcome (Class): success = 1, failure = 0

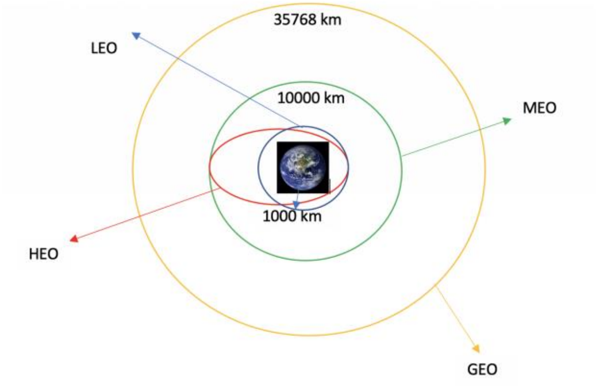


- At all 3 sites, the majority of launches had payload masses below 7,000 kg.
- Most successful launches occurred in this range, although successes tended to be in a lower range at VAFB SLC-4E than at KSC LC-39A.
- Heavy payloads: VAFB SLC-4E did not launch any rockets with heavy payload masses (over 10,000 kg). The other 2 sites each had one successful launch in the heaviest range (14,000-16,000 kg).

What is the launch success rate for each orbit type?



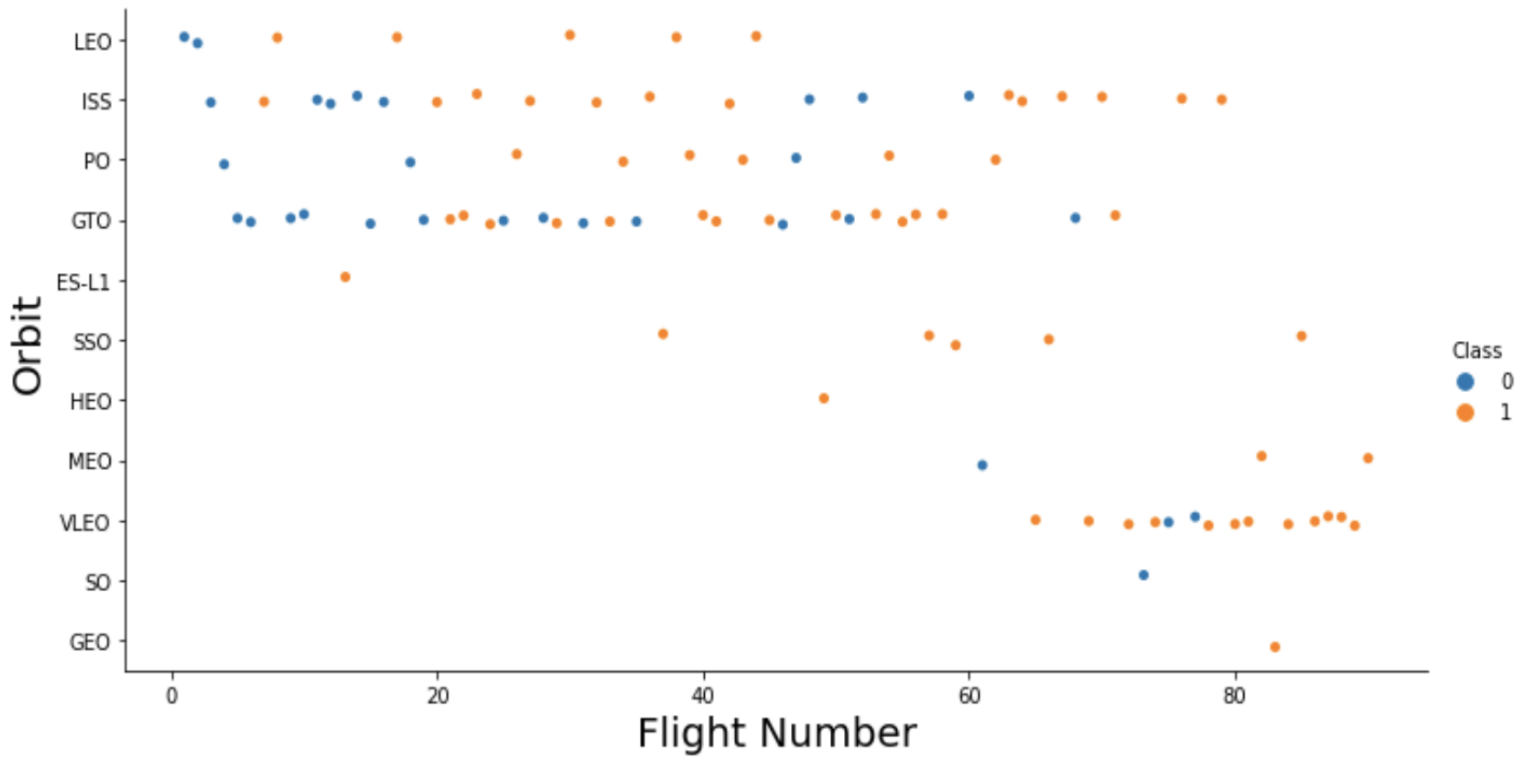
Depiction of LEO, MEO, GEO, & HEO Orbits:



- Orbits with high success rates: launches with SSO, ES-L1, GEO, and HEO orbits all had 100% success rates
- Orbits with low success rates: launches with GTO, ISS, PO, and MEO orbits had success rates from 55-65%.

How do Flight Number and Orbit Type relate to Launch Outcome?

Flight Number: total launch attempts for rocket
Launch Outcome (Class): success = 1, failure = 0

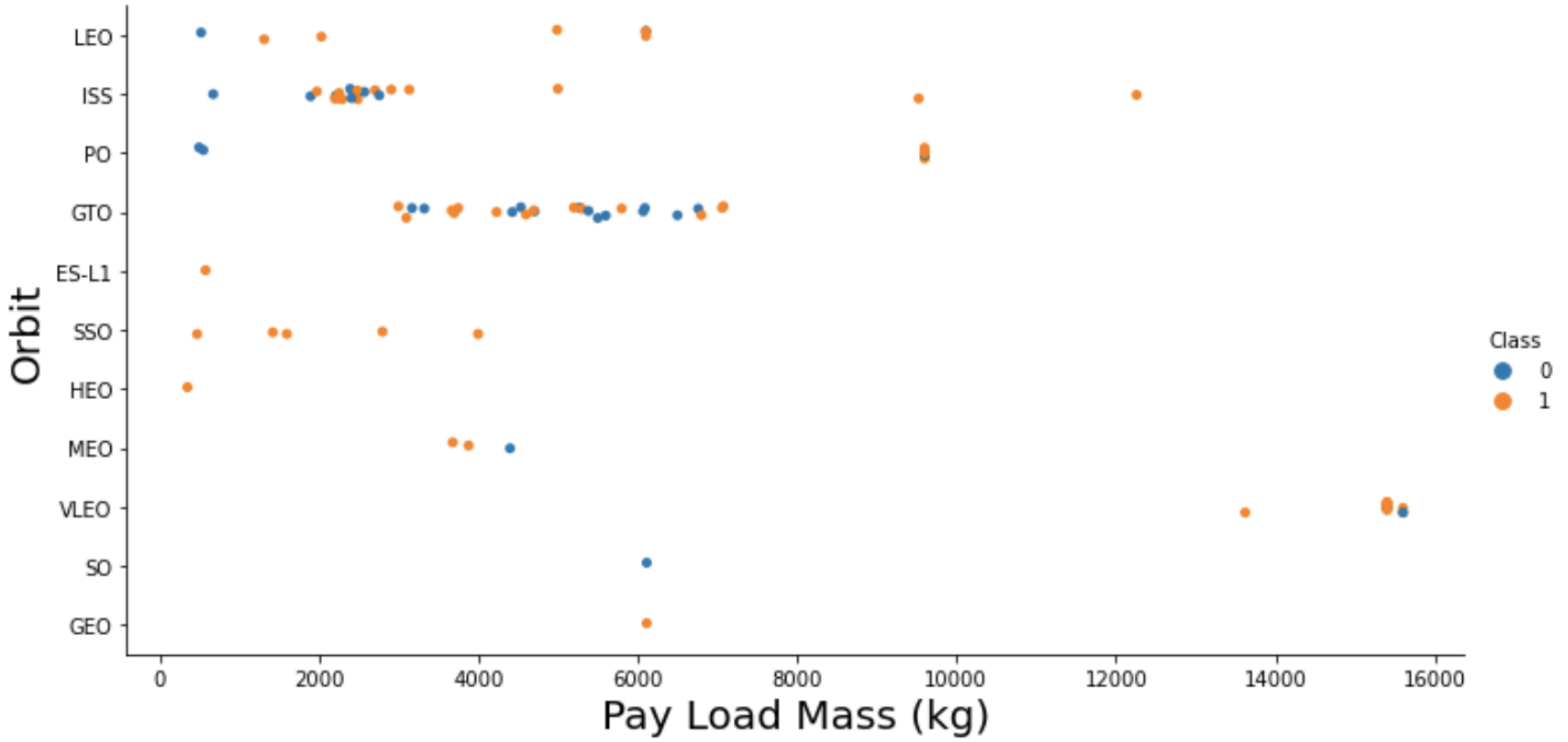


- In LEO, ISS, PO, and GTO orbits, most launches with lower flight numbers were successful (with some exceptions over 40).
- In the GTO orbit, the overall relationship between flight number and success was unclear.
- SSO, VLEO, and MEO orbits tended to have higher flight numbers and fewer successes.
- Some orbit types (e.g., HEO) had only a single launch attempt, making it difficult to determine a pattern with flight number or success.

How do Payload Mass and Orbit Type relate to Launch Outcome?

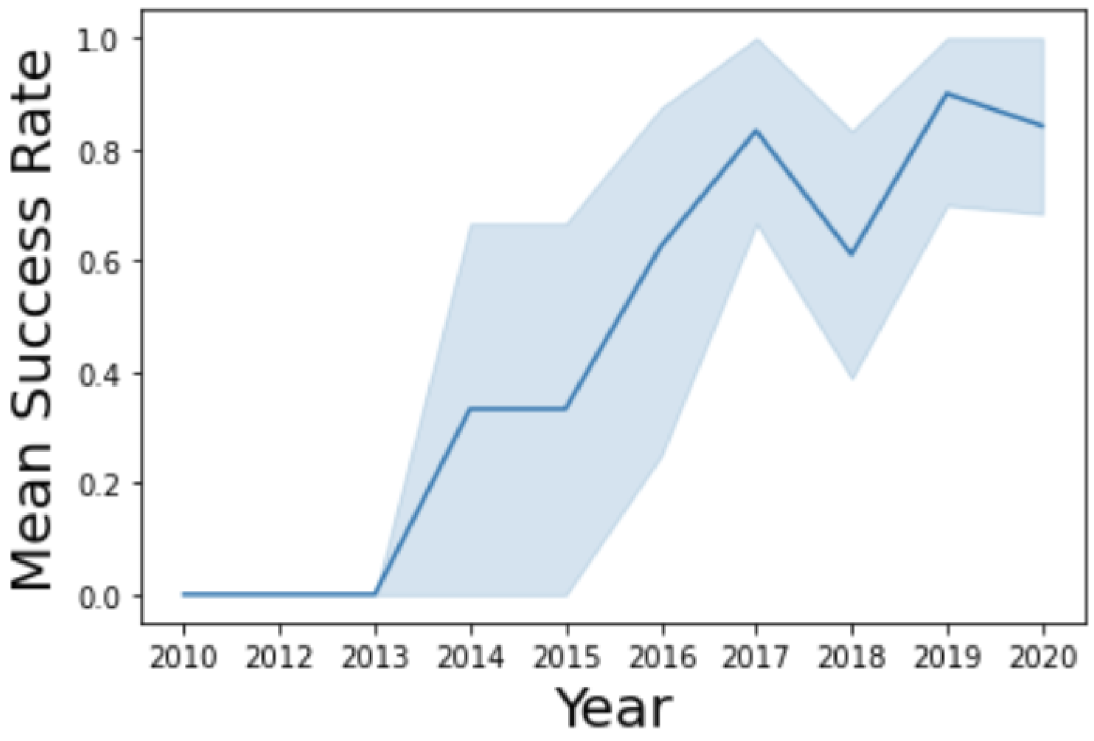
Payload Mass: mass transported by rocket (in kg)

Launch Outcome (Class): success = 1, failure = 0



- At lighter payloads, ISS and GTO orbits appeared to have higher rates of successful landings (but also more attempts and therefore more data points).
- Only the GTO, MEO, and GEO orbits had successful launches with payloads of 6,000 kg or greater.

How has the average launch success rate per year changed over time?



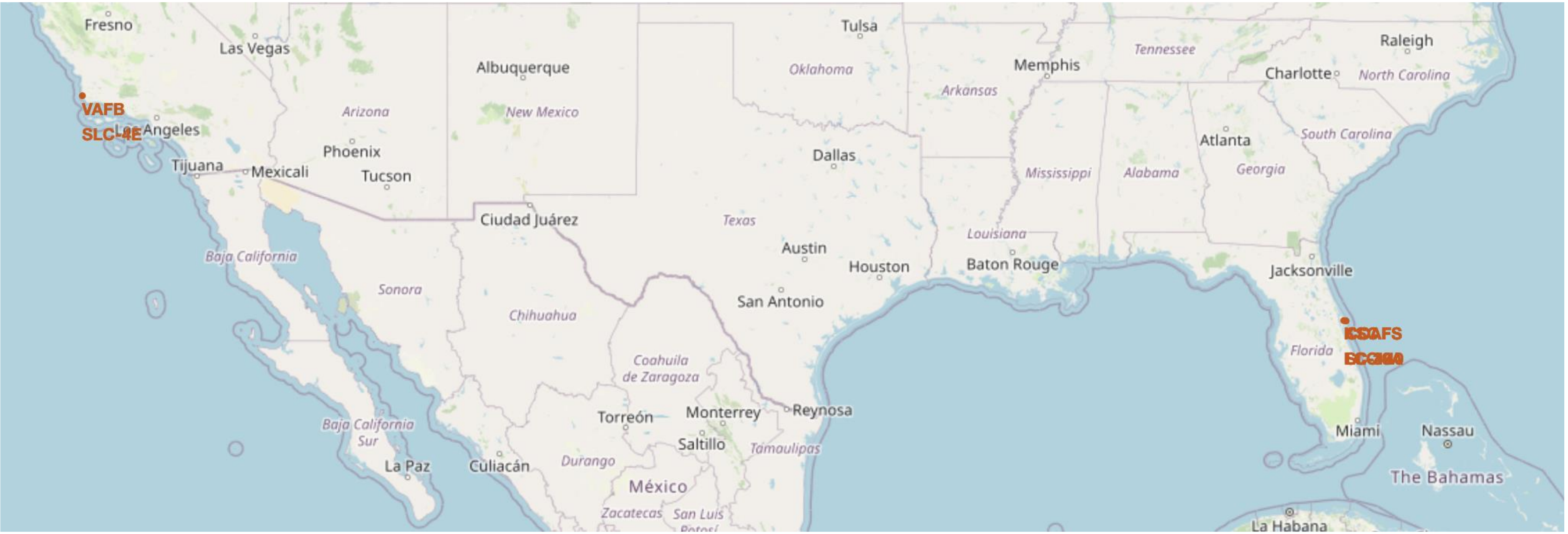
- Overall, mean annual success rates increased dramatically between 2010 and 2020.
- Average success rates per year rose dramatically between 2013-2014, 2015-2017, and 2018-2019.
- Success rates plateaued in 2015, declined sharply in 2018, and decreased slightly again in 2020.

Results:

Launch Site Locations (Folium Map)

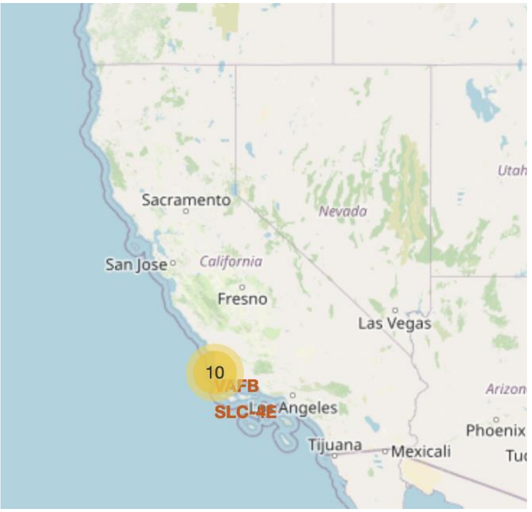
Launch Site Locations

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610746

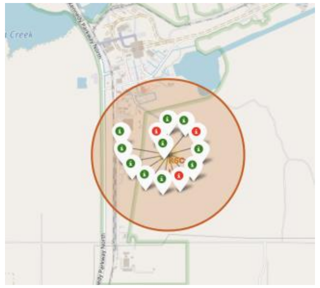


- All four launch sites are located near coastlines in Southern regions of the U.S. (relatively nearer to the Equator)

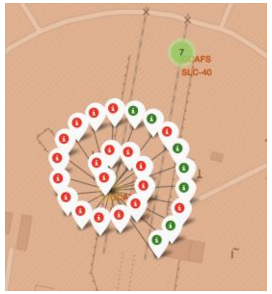
Which launch sites have relatively high success rates?



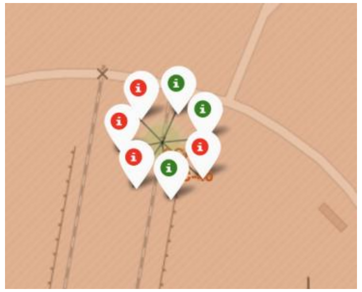
KSC-LC 39A



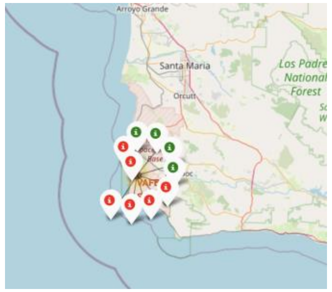
CCAFS-LC 40



CCAFS-SLC 40

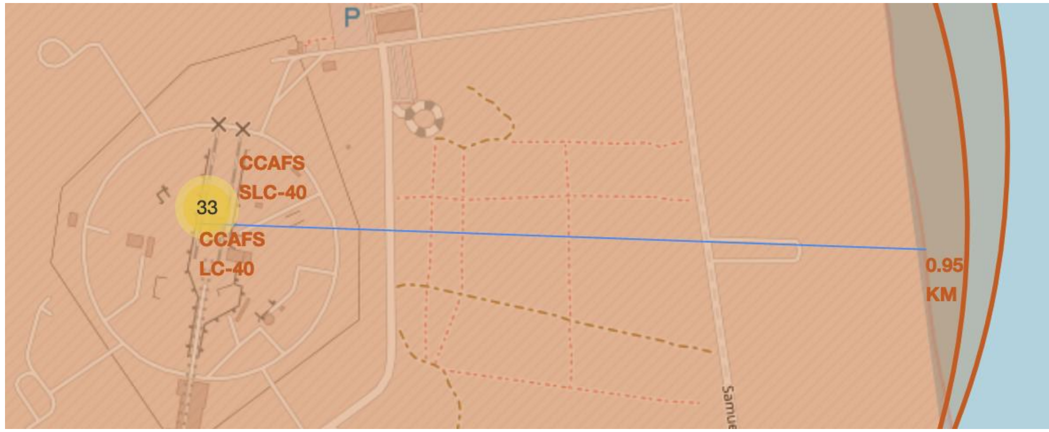


VAFB-SLC 4E



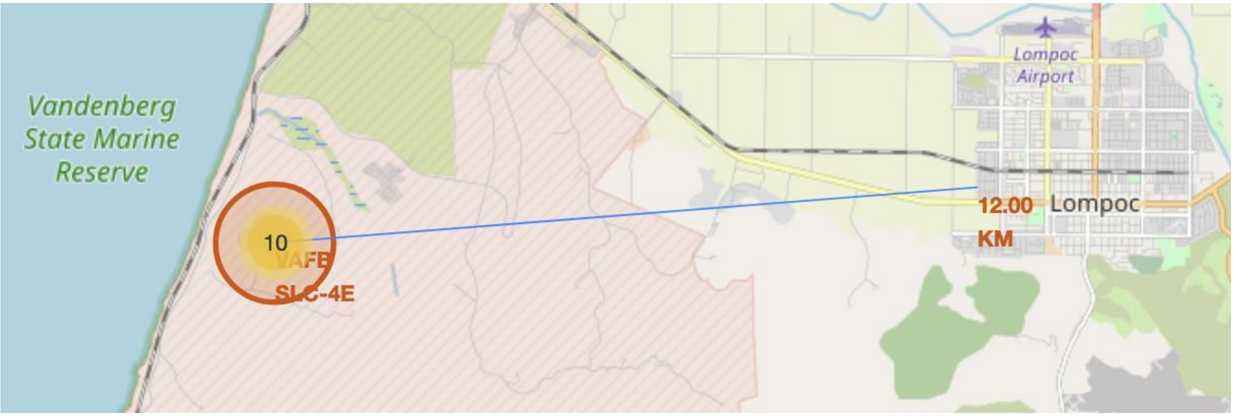
- KSC-LC 39A appears to have a high success rate compared to other sites, while CCAFS-SLC 40 and VAFB-SLC 4E are more moderate. CCAFS-LC 40 seems to have the lowest rate of successful launches.

Distances between launch sites and coastlines



- CCAFS LC-40 and CCAFS SLC-40 are approximately 0.96 KM from the Atlantic coastline in Florida, while VAFB is about 1.26 KM from the Pacific Coastline.
- The close proximity of all 4 launch sites to nearby coastlines makes sense given the need to land rockets in open water

Distances between launch sites and nearby cities



- CCAFS LC-40 and its neighboring sites are roughly 18.1 KM from the nearest city in Florida (Capa Canaveral)
- VAFB SLC-4E is about 12 KM from Lompoc (CA)
- Launch sites may be located slightly further from cities than from coastlines or other proximities due to safety and security concerns

Results:
Predictive Analysis (ML Classification)

Model 1: Logistic Regression

```
# Create a logistic regression object then create a GridSearchCV object logreg_cv with cv = 10.
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

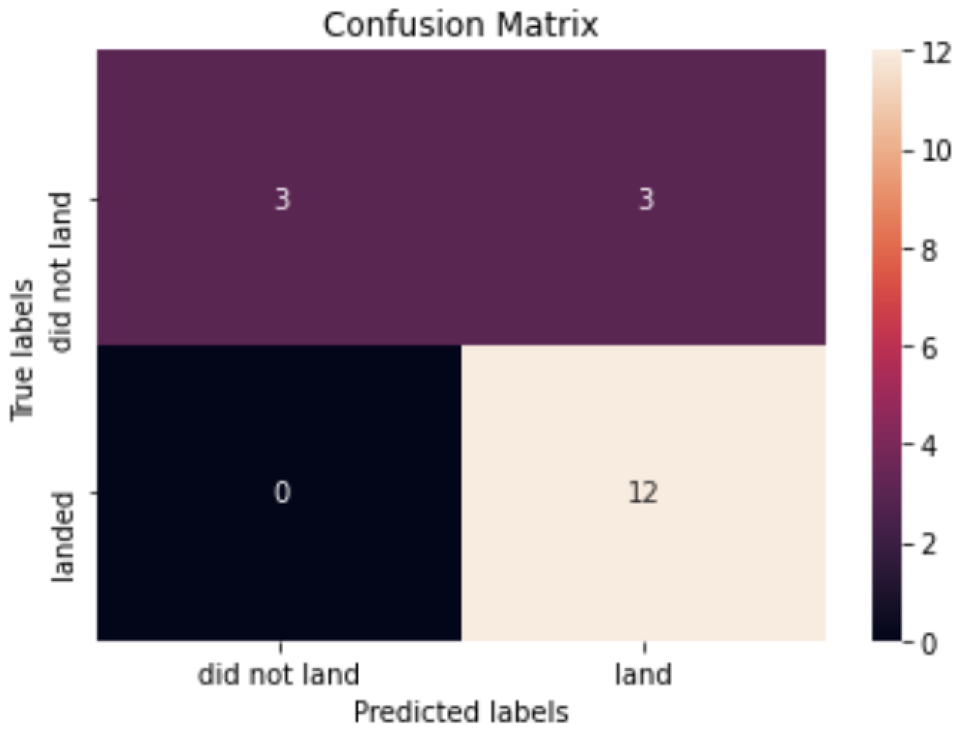
```
# Fit the object to find the best parameters from the dictionary parameters.
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()
logreg_cv = GridSearchCV(lr,parameters,cv=10)
```

```
# output the GridSearchCV object for logistic regression
logreg_cv.fit(X_train,Y_train)
print("tuned hyperparameters :(best parameters) ",logreg_cv.best_params_) # display the best parameters
print("accuracy :",logreg_cv.best_score_) # display accuracy on the validation data
```

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

```
print('Accuracy=  ',logreg_cv.score(X_test,Y_test))
```

```
Accuracy=   0.8333333333333334
```



- The logistic regression model's predictions were 83.33% using the test data.
- The confusion matrix shows us that this method can distinguish between landing outcome classes, but false positives are a major issues (3 launches were incorrectly predicted to land successfully).

Model 2: SVM (support vector machine)

```
# Create a support vector machine object then create a GridSearchCV object svm_cv with cv - 10.
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
```

```
# Fit the object to find the best parameters from the dictionary parameters.
svm_cv = GridSearchCV(svm,parameters,cv=10) # same w/o accuracy scoring
svm_cv.fit(X_train,Y_train)
```

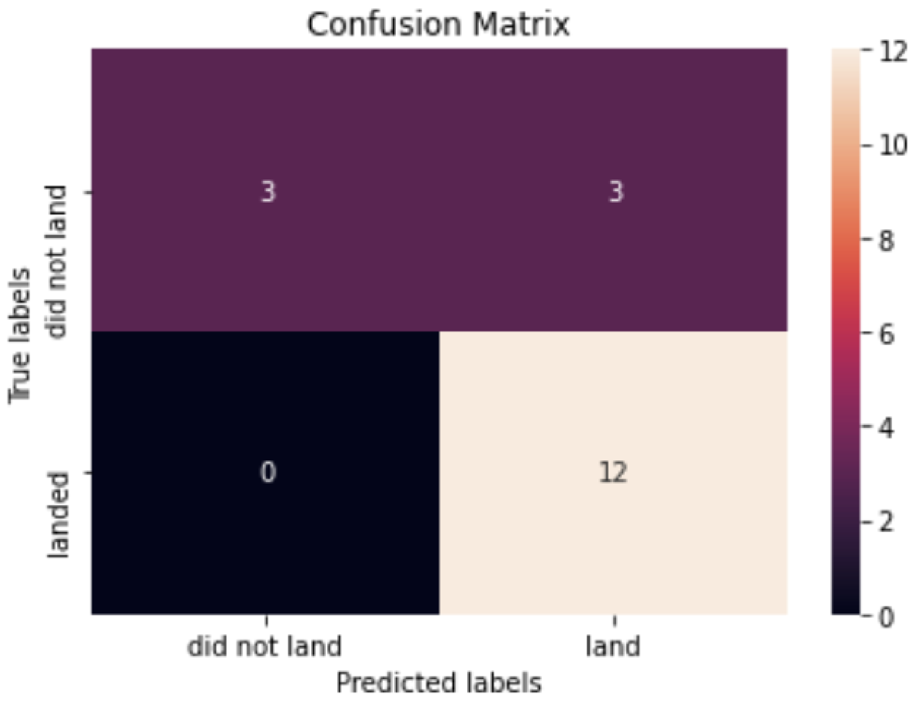
```
5]: GridSearchCV(cv=10, estimator=SVC(),
             param_grid={'C': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
       1.00000000e+03]),
                         'gamma': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
       1.00000000e+03]),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

```
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

```
print("accuracy: ",svm_cv.score(X_test,Y_test))
```

```
accuracy:  0.8333333333333334
```



- As in logistic regression, the SVM prediction accuracy rate is 83.33%.
- The confusion matrix demonstrates that this method also had issues with false positives (launches incorrectly predicted to land successfully).

Model 3: Decision Tree

```
# Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10.
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()
```

```
# Fit the object to find the best parameters from the dictionary parameters.
tree_cv = GridSearchCV(tree,parameters,cv=10)
tree_cv.fit(X_train,Y_train)
```

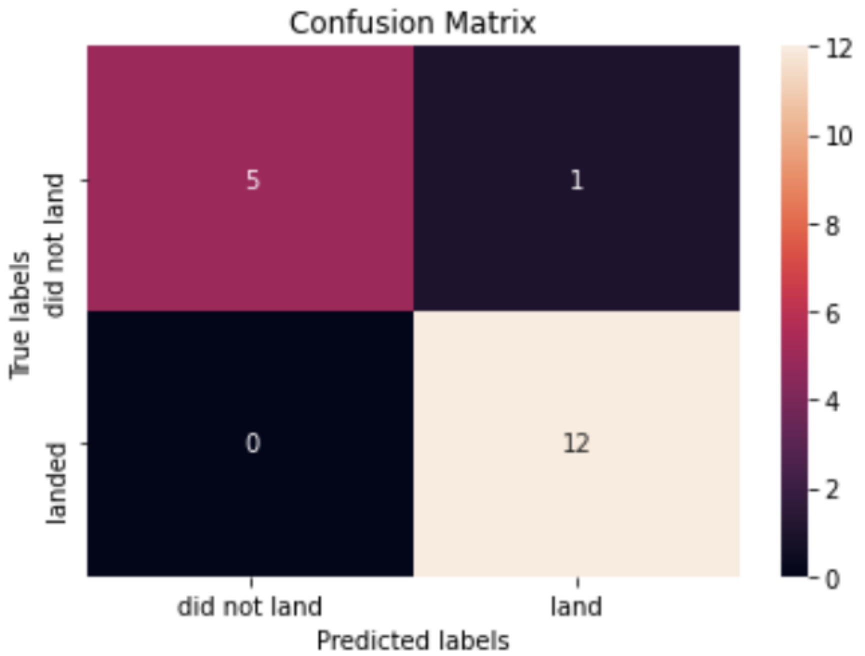
```
4]: GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
                 param_grid={'criterion': ['gini', 'entropy'],
                             'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                             'max_features': ['auto', 'sqrt'],
                             'min_samples_leaf': [1, 2, 4],
                             'min_samples_split': [2, 5, 10],
                             'splitter': ['best', 'random']})
```

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 12, 'max_features': 'auto',
0, 'splitter': 'best'}
accuracy : 0.875
```

```
print("accuracy: ",tree_cv.score(X_test,Y_test))
```

```
accuracy:  0.9444444444444444
```



- The decision tree model has a prediction accuracy rate of **94.44%** – much higher than in the two previous models.
- The confusion matrix features only 1 false positive, indicating that this method is more accurately distinguishing between outcome classes.

Model 4: K Nearest Neighbors (KNN)

```
# Create a k nearest neighbors object then create a GridSearchCV object knn_cv with cv = 10.
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()
```

```
#Fit the object to find the best parameters from the dictionary parameters.
knn_cv = GridSearchCV(KNN,parameters,cv=10)
knn_cv.fit(X_train,Y_train)
```

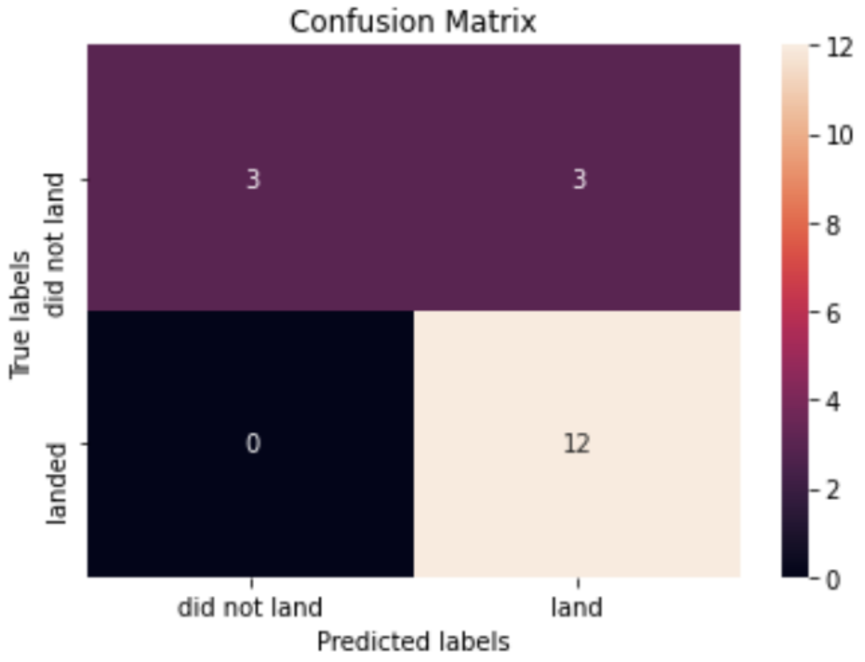
```
8]: GridSearchCV(cv=10, estimator=KNeighborsClassifier(),
             param_grid={'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                         'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
                         'p': [1, 2]})
```

```
print("tuned hpyerparameters :(best parameters) ",knn_cv.best_params_)
print("accuracy :",knn_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858
```

```
print("accuracy: ",knn_cv.score(X_test,Y_test))
```

```
accuracy:  0.8333333333333334
```



- The KNN method had a **83.33%** rate of prediction accuracy, on par with the SVM and logistic regression methods.
- The confusion matrix for this method also indicates false positive issues.

Best performing method

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8892857142857142
Best Params is : {'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'random'}
```

Which method had the highest accuracy in predicting launch outcomes?

- The decision tree model performed best out of the four methods, with an accuracy score of 0.8893.
- The KNN & SVM methods both had accuracy scores of 0.8492.
- The logistic regression method had the lowest accuracy score of 0.8464.

INSIGHTS & CONCLUSION

Question 1: What key features of the Falcon 9 launches are important for predicting launch outcomes?

Key variables related to Falcon 9 launch outcomes include launch site location, flight number, payload mass, orbit type, and booster version.

- Launch successes varied by site location: CCAFS LC-40 had a lower average success rates (60%), while KSC LC-39A and VAFB SLC 4E had higher rates (77%).
- First stage launches tended to be more successful at higher flight numbers and lower payload masses.
    - Most successful launches occurred in payload range of 7,000 kg or lower, but this varied somewhat by site
    - Typical flight numbers varied by site
- SSO, ES-L1, GEO, and HEO orbit launches had 100% average success rates, while GTO, ISS, PO, and MEO orbit launches had much lower average success rates (55-65%.)
    - The overall relationship between orbit type and flight number was less consistently clear.
- Specific Falcon 9 booster versions may be used depending on the launch's landing type (e.g., drone ship) or payload mass.
- Overall, average launch success rates per year increased dramatically between 2010 and 2020. Most recently, there were sharp declines in 2020.

Question 2: Based on findings from EDA, which ML classification model can most accurately predict Falcon 9 launch outcomes?

Of the four classification methods used with the SpaceX launch data, the decision tree model featured the most accurate predictions of launch successes and failures

All models had some issues with false positive predictions, or incorrectly predicting launches that would successfully land. As this type of prediction error is crucial to avoid given the business goals, it would be useful to explore the characteristics of these launches in order to improve the model.

As many factors impact landing outcomes and success rates may also vary based on the specific combinations of launch features (payload, orbit, booster version, site, etc.), the accuracy of the predictions could also benefit from integrating additional detail or data sources into the predictive model.