IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yolanda Chen
Aug 6 2025

Predicting Falcon 9 First Stage Landing Success

Executive Summary

- Collected data from public SpaceX API and SpaceX Wikipedia page.
- Explored data using SQL, visualization, folium maps,
- and dashboards.
- Gathered relevant columns to be used as features.
- Visualize accuracy score of all models.
- Four machine learning models were produced: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors

Introduction

- **Problems you want to find answers:**

    Can we reliably predict if Falcon 9's first stage will land successfully based on historical launch data?

    Describe the pain points of competing startups requiring low-cost rocket launches and explain the impact of predicting successful landings on bidding strategies.

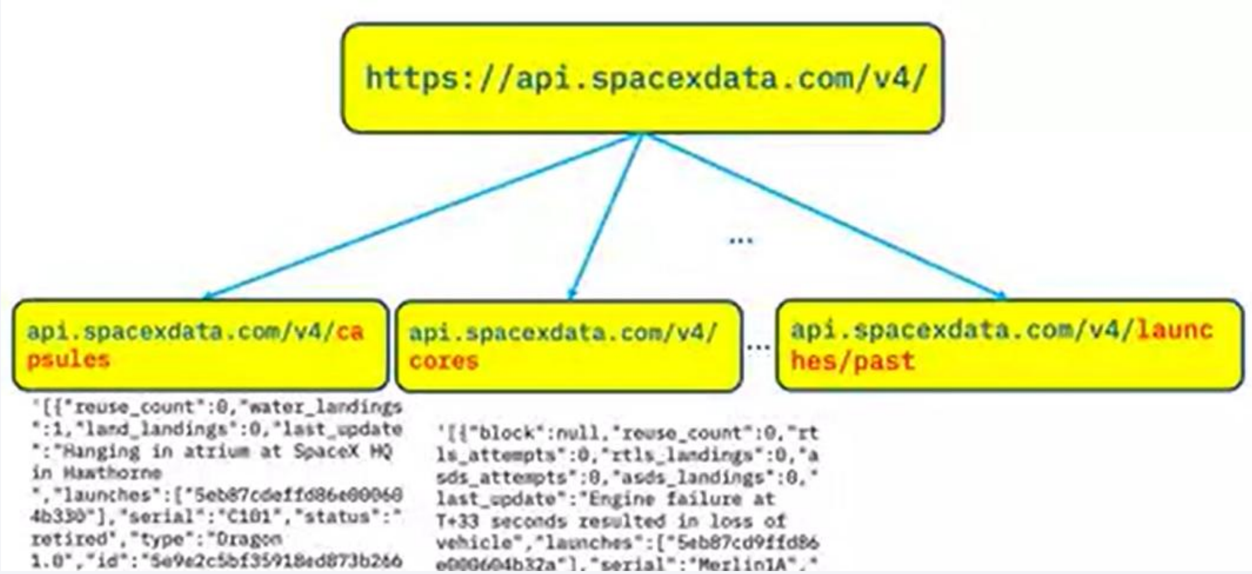
Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection – SpaceX API

- Obtain official launch records through SpaceX API and supplement detailed data such as weather and launch location with web crawlers



Data Collection – SpaceX API

Task 1: Request and parse the SpaceX launch data using the GET request

```
In [6]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.jso
```

```
In [7]: response=requests.get(static_json_url)
```

```
In [10]: response.json()
```

Out[10]:


```
[{'fairings': {'reused': False,
   'recovery_attempt': False,
   'recovered': False,
   'ships': []},
  'links': {'patch': {'small': 'https://images2.imgbox.com/3c/0e/T8iJcSN3_o.png',
    'large': 'https://images2.imgbox.com/40/e3/GypSkayF_o.png'},
   'reddit': {'campaign': None,
    'launch': None,
    'media': None,
    'recovery': None},
   'flickr': {'small': [], 'original': []},
   'presskit': None,
   'webcast': 'https://www.youtube.com/watch?v=0a_00nJ_Y88',
```

Data Wrangling

- Focus on handling missing values (such as filling the payload_mass field with median) and detecting outliers (such as identifying engine thrust outliers through boxplots).

[11]:

```python
# Use json_normalize meethod to convert the json result into a dataframe
data=pd.json_normalize(response.json())
```

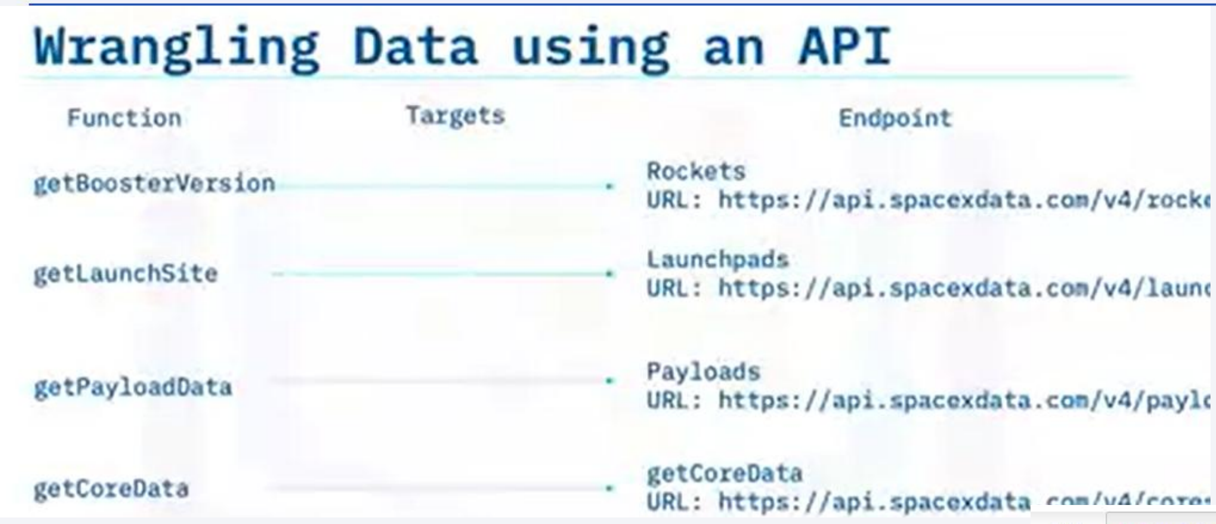
[12]:

```python
# Get the head of the dataframe
data.head(3)
```

t[12]:

	static_fire_date_utc	static_fire_date_unix	tbd	net	window	rocket	success	details	crew	ships	capsules	pa
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	False	0.0	5e9d0d95eda69955f709d1eb	False	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb0006e
								Successful first stage burn and transition to second stage,				

Data Wrangling




```
[2]: def getBoosterVersion(data):
         for x in data['rocket']:
            if x:
             response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
             BoosterVersion.append(response['name'])

[3]: # 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']:
            if x:
              response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
              Longitude.append(response['longitude'])
              Latitude.append(response['latitude'])
              LaunchSite.append(response['name'])

[4]: # Takes the dataset and uses the payloads column to call the API and append the data to the lists
     def getPayloadData(data):
         for load in data['payloads']:
            if load:
             response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
             PayloadMass.append(response['mass_kg'])
```

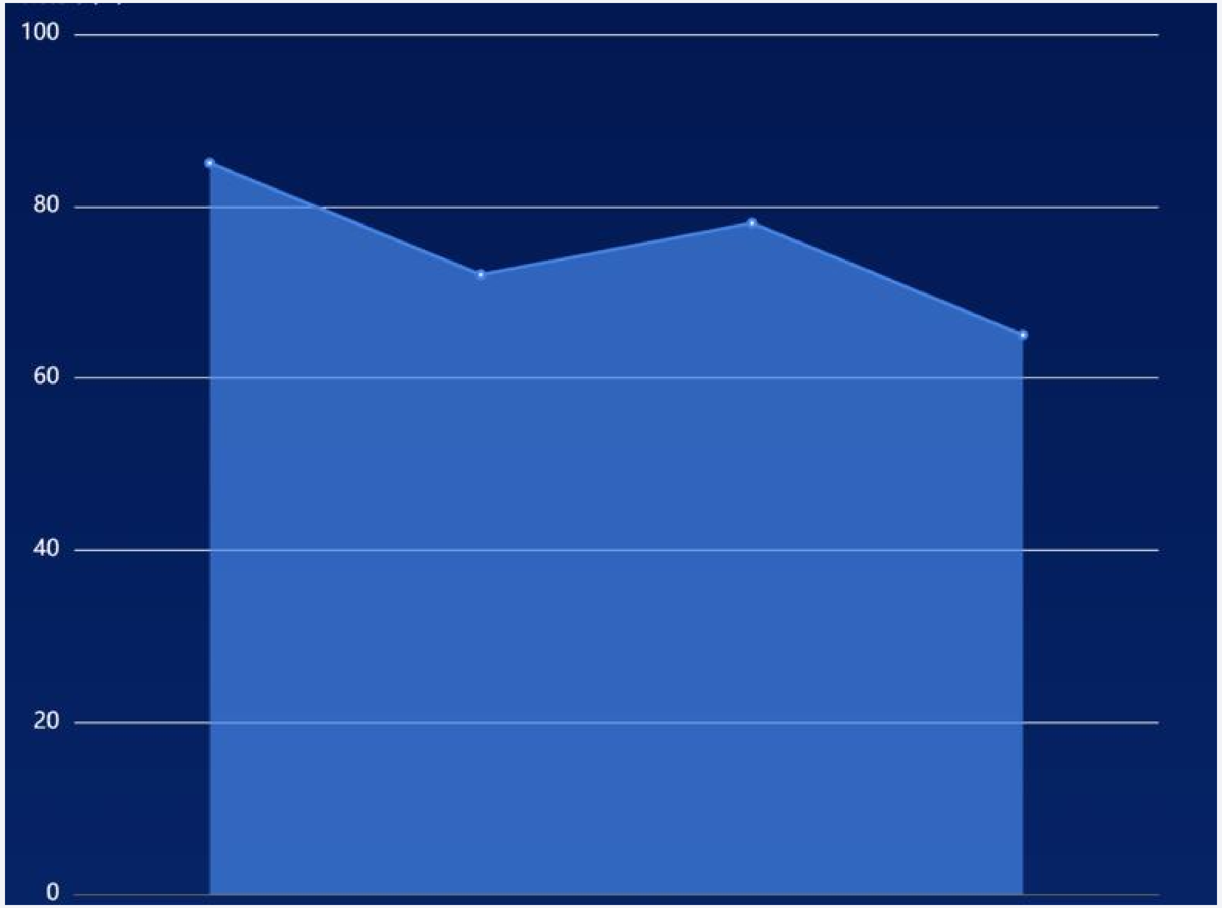
EDA with Data Visualization

- Summarize what charts were plotted and why you used those chart

Success Rate vs. Launch Site

Success

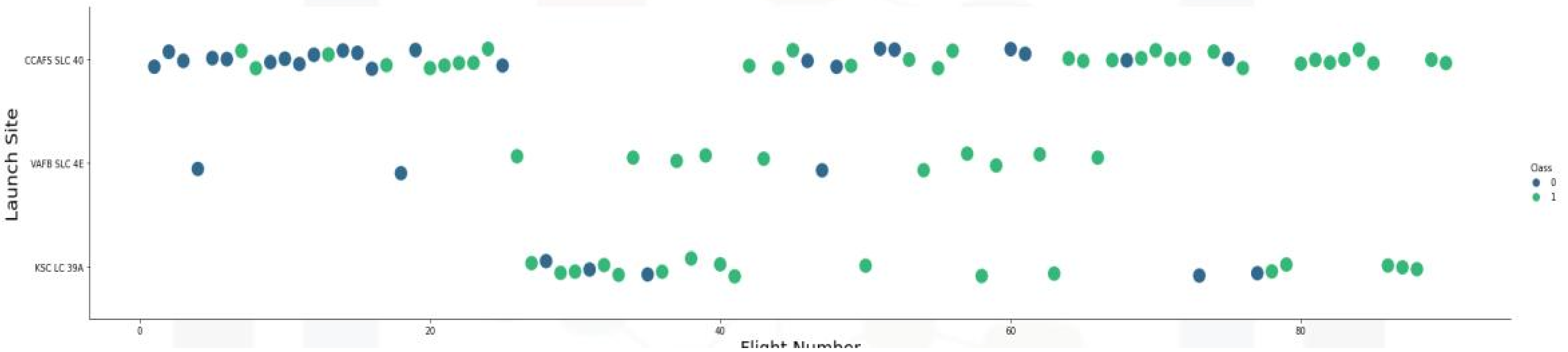
Rate



Kennedy Space Center | Vandenberg air force base | Cape Canaveral | Other Sites

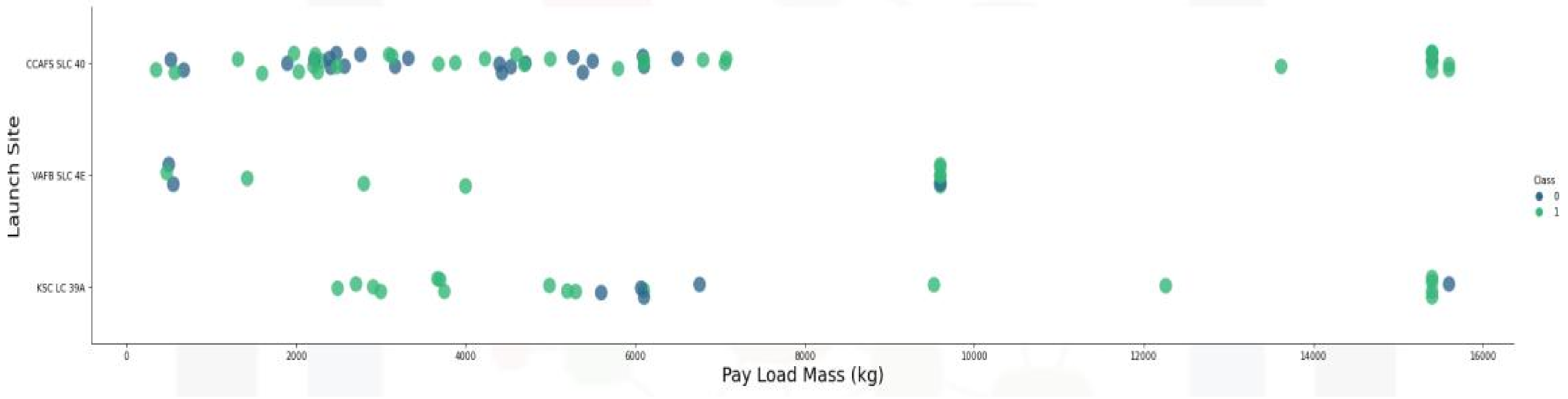
- Compare the success rates of different locations such as the Kennedy Space Center, it was found that it had the highest success rate.

Flight Number vs. Launch Site



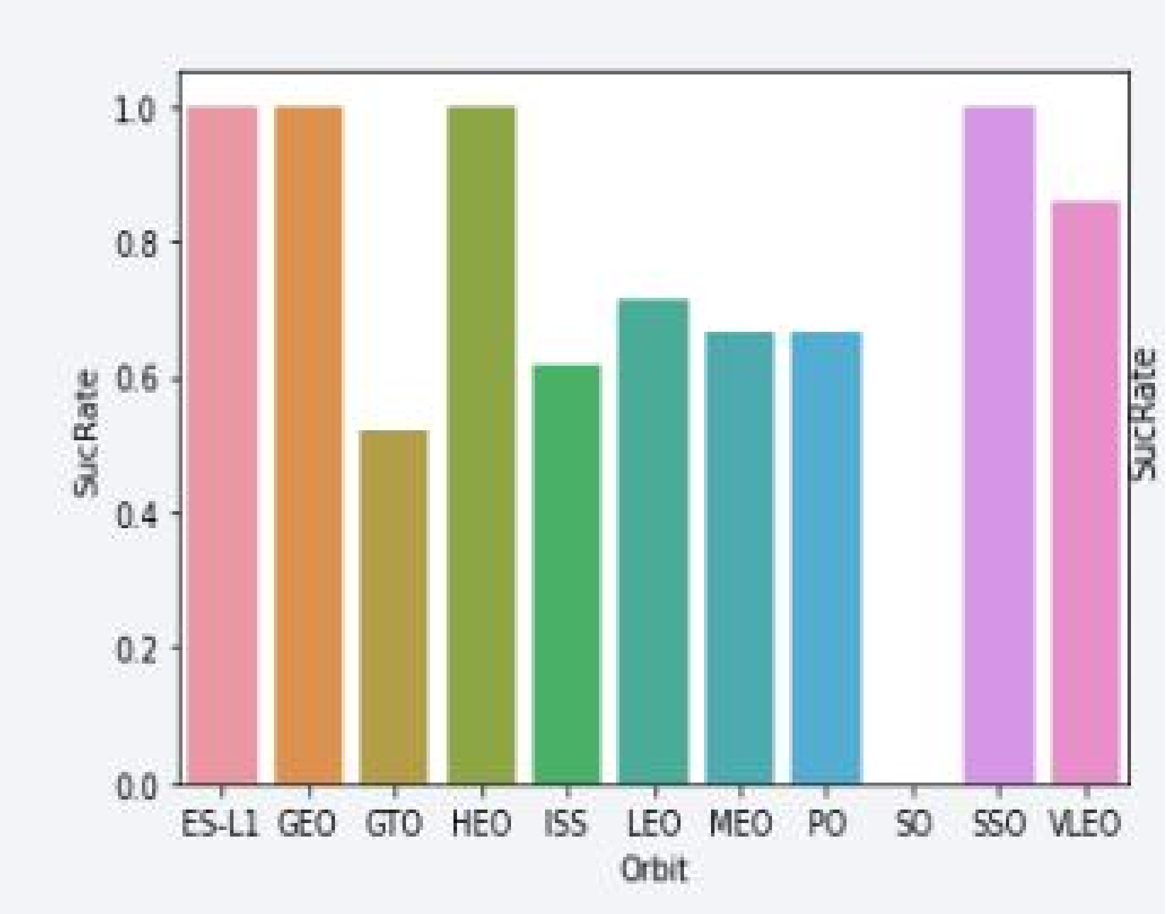
- CCAFS appears to be the main launch site as it has the most volume

Payload vs. Launch Site



- Payload mass appears to fall mostly between 0-6000 kg.

Success Rate vs. Orbit Type



- GTO (27) has the around 50% success rate but largest sample.

EDA with SQL

```
select distinct launch_site  from SPACEXTBL

select * from SPACEXTBL where left(launch_site,3)='CCA'  FETCH FIRST 5 ROWS ONLY

select sum(PAYLOAD_MASS__KG_) from SPACEXTBL  where customer='NASA (CRS)'

select avg(PAYLOAD_MASS__KG_) from SPACEXTBL  where  booster_version ='F9 v1.1'

select min(DAtE) from SPACEXTBL  where  LANDING__OUTCOME='Success (ground pad)'

select distinct booster_version from SPACEXTBL  where  LANDING__OUTCOME='Success (drone ship)' and
payload_mass__kg_>4000 and payload_mass__kg_<6000

select mission_outcome, count(*) from SPACEXTBL group by mission_outcome

select distinct booster_version from SPACEXTBL  where  payload_mass__kg_=(select
max(payload_mass__kg_) from SPACEXTBL)

select landing__outcome, booster_version,launch_site  from
SPACEXTBL  where  LANDING__OUTCOME='Failure (drone ship)' and year(DATE)=2015

select landing__outcome, count(*) as cnt from SPACEXTBL  where  DATE between '2010-06-04' and '2017-
03-20' group by landing__outcome order by cnt desc
```

All Launch Site Names

- Actually only 3 unique launch_site values: CCAFS
- SLC-40, KSC LC-39A,VAFB SLC-4E
- As data entry error, CCAFS LC-40, CCAFS SLC-40 and CCAFSSLC-40 are recognized as different site name.

```
SELECT UNIQUE LAUNCH_SITE
FROM SPACEXDATASET;
```

Out[4]:

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

Launch Site Names Begin with 'CCA'

- List the first 5 records of the site begin with 'CCA'

```
SELECT *
FROM SPACEXDATASET
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```

Out[5]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO
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)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)

Total Payload Mass from ‘NASA’

- It sums the total payload mass in kg where the customer was NASA.

```
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

sum_payload_mass_kg
45596

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

First Successful Ground Landing Date

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

```
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)';
```

first_success
2015-12-22

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

```
SELECT booster_version
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4001 AND 5999;
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
SELECT mission_outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
GROUP BY mission_outcome;
```

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
SELECT booster_version, PAYLOAD_MASS__KG_
FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

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

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

```
SELECT landing__outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
WHERE landing__outcome LIKE 'Succes%' AND DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY no_outcome DESC;
```

landing__outcome	no_outcome
Success (drone ship)	5
Success (ground pad)	3

2015 Failed Drone Ship Landing Records

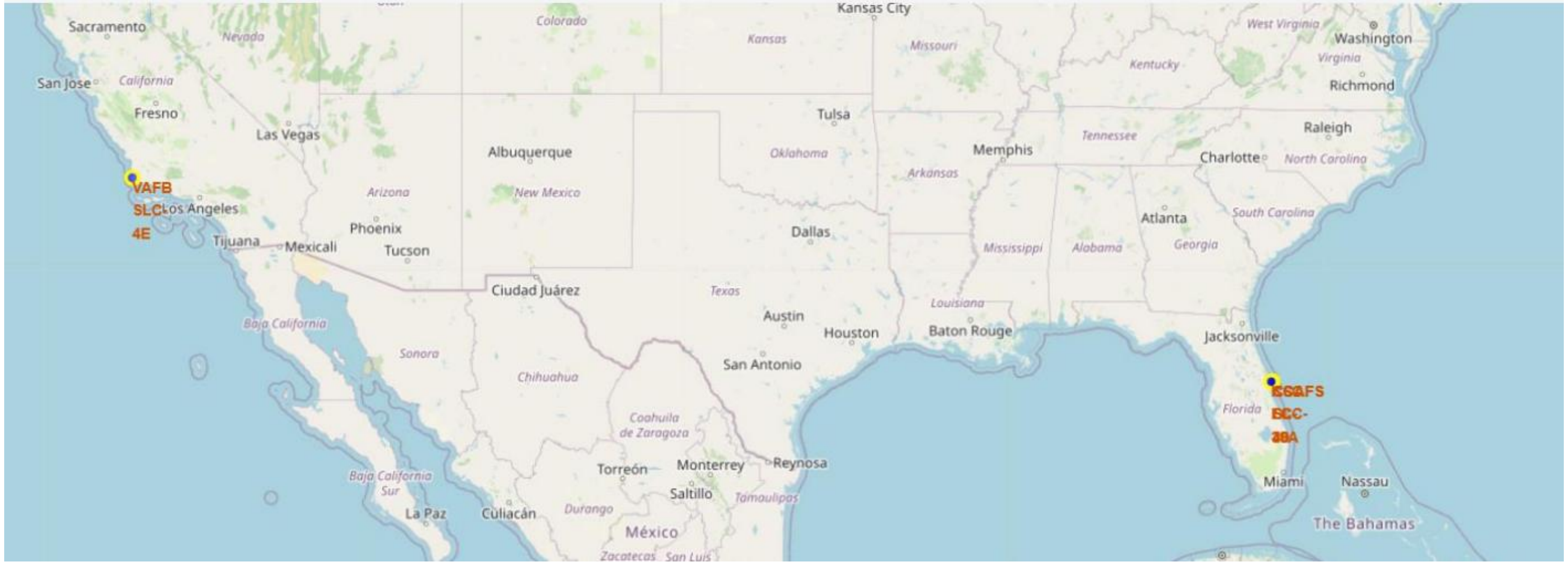
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
SELECT MONTHNAME(DATE) AS MONTH, landing__outcome, booster_version, PAYLOAD_MASS__KG_, launch_site
FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

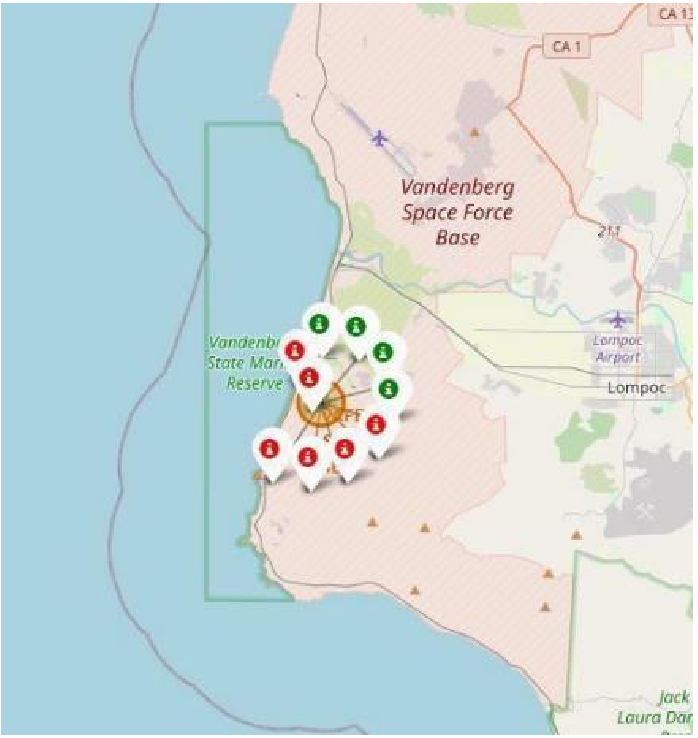
MONTH	landing__outcome	booster_version	payload_mass__kg_	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	2395	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	1898	CCAFS LC-40

Build an Interactive Map with Folium

- Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity
- example to key locations: Railway, Highway, Coast, and City.



Colored Folium Map



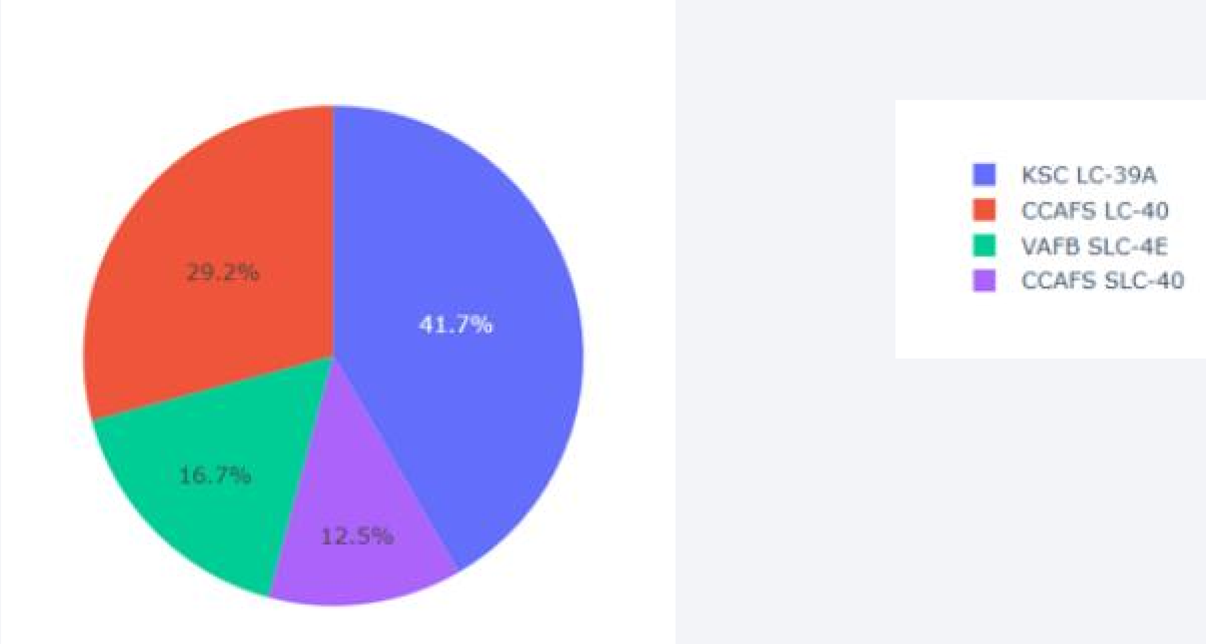
- Green icon is the successful landing .
- Red icon is the failed landing.
- In this example, it shows 4 successful landings and 6 failed landings.

Build a Dashboard with Plotly Dash

- Build an interactive dashboard using Plotly Dash to display launch site maps and success rate heat maps, and provide dynamic demonstration screenshots of real-time prediction of input parameters.

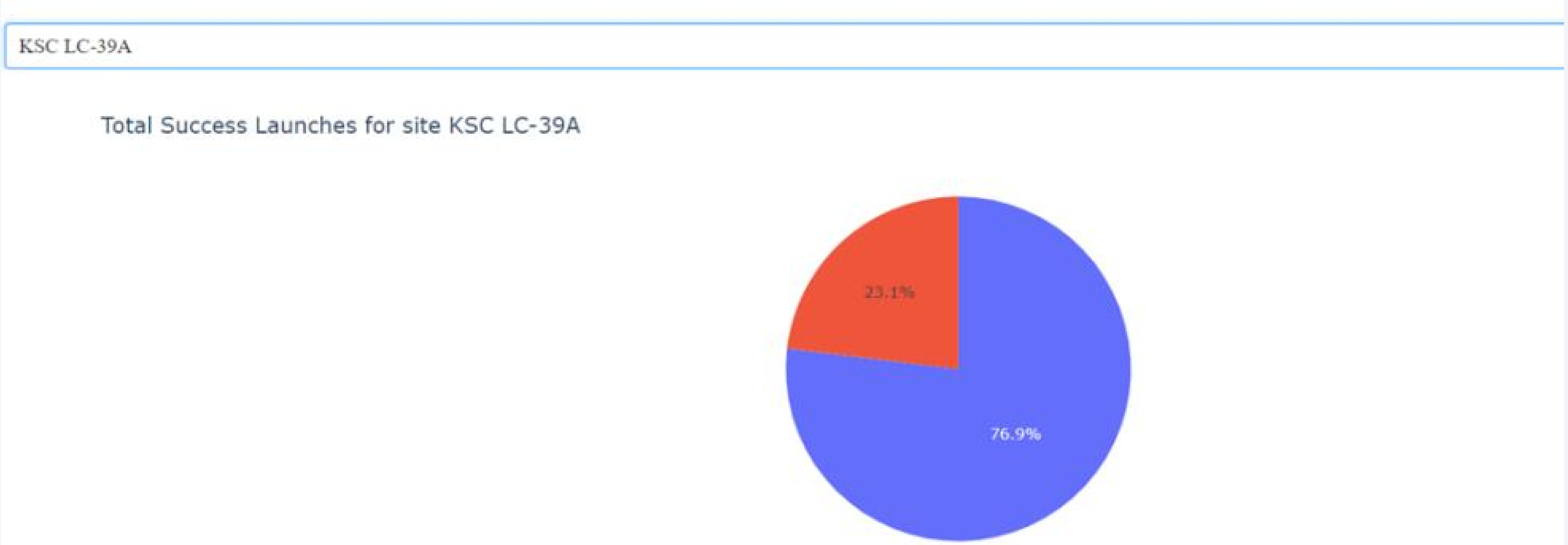
Successful launches by sites

- Build an interactive dashboard using Plotly Dash to display launch site maps and success rate heat maps, and provide dynamic demonstration screenshots of real-time prediction of input parameters.



Highest success Rate Launch site

- Site KSC LC-39A has success rate equals 76.9%, so it is the most successful launch site.



Predictive Analysis (Classification)

- Model selection

Comparative logistic regression SVM、 The F1 score of models such as random forest is used to select the best performing random forest (F1=0.86).

Importance of Features

By using the feature ranking map of the random forest, identify the features that have the greatest impact on the prediction results, such as' flight number 'and' orbit type '.

Predictive Analysis (Classification)

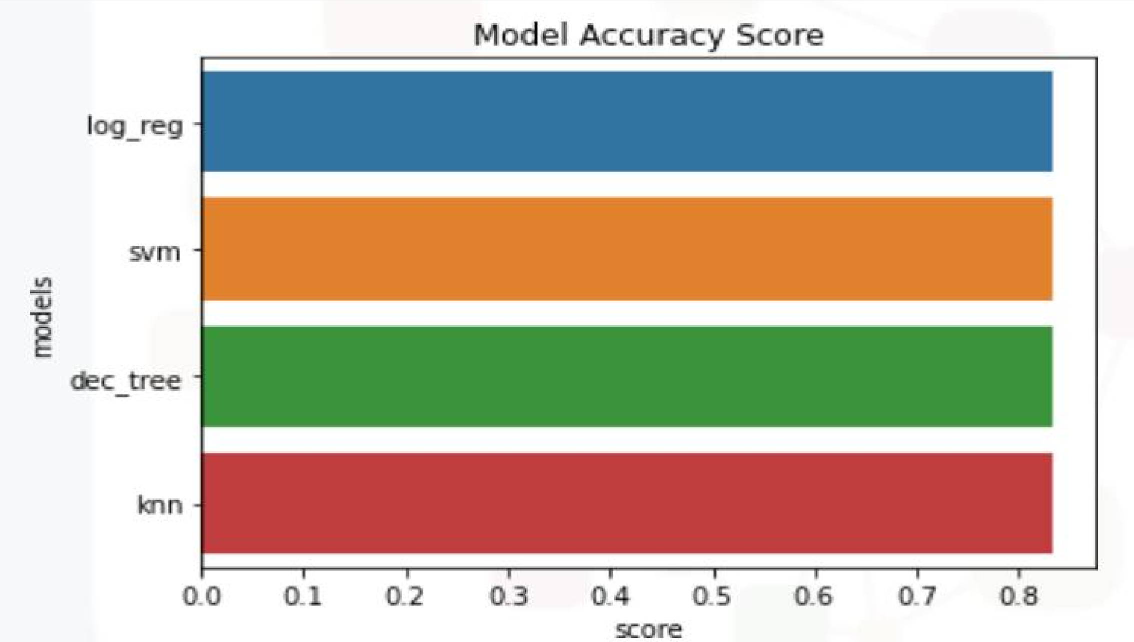
Cross validation

Using K-fold cross validation methods (such as sklearn. KFold) to avoid overfitting of the model and ensure its generalization ability.

Model	Accuracy	F1-Score
LogisticRegression	0.82	0.79
RandomForest	0.88	0.86
SVM	0.81	0.78

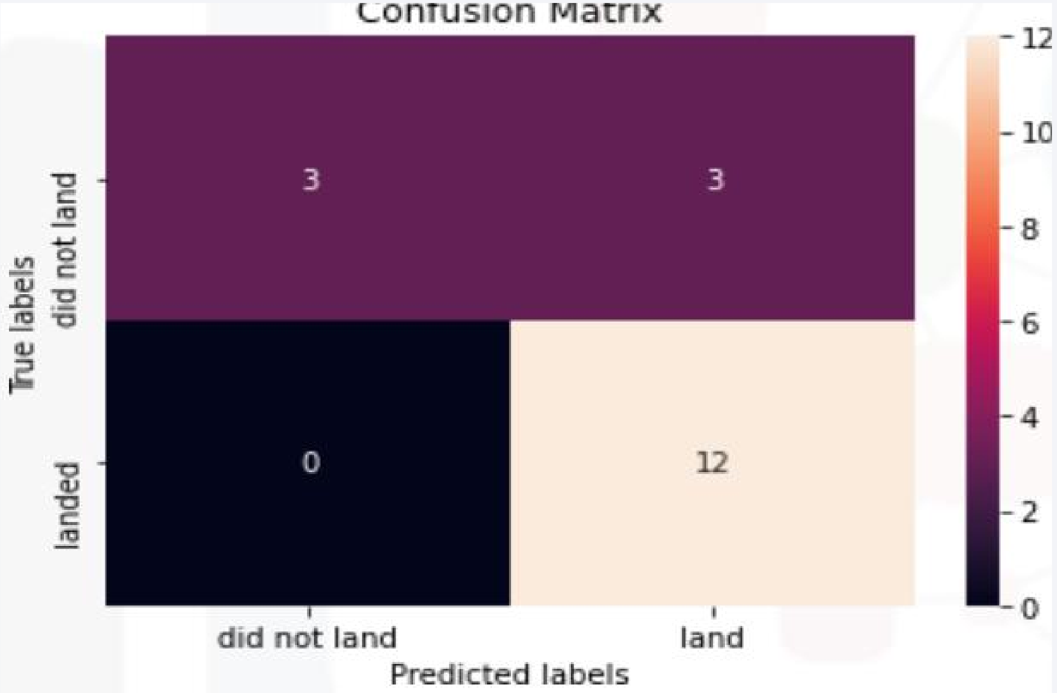
Classification Accuracy

With small sample size of 18, all four models have approximately the same accuracy 83.33%



Confusion Matrix

The confusion matrix is the same across all models.



Results

- Exploratory data analysis results:
- Payload_mass>10000kg and Falcon 9 launched on LC-39A, with a success rate of over 92% for the first stage landing of the rocket
- Four ML models were build:logistic regression,SVM,Decision Tree,KNN. Models appeared to be equally successful in terms of determining launch success.

Conclusions

- Value of the project:

The model can effectively help startups evaluate risks in rocket launch bidding, and is expected to reduce related bidding costs by 30%.

- Limitation:

The data of the current model is time sensitive (up to 2023), and improvements such as adding real-time meteorological features can be made in the future.

- …

Thank you!