Data Science Capstone Project

<Oluwatobi Ogunsanya>

<20/08/21>

Outline

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

Executive Summary

- I collected data from various sources. After the raw data was collected, I needed to improve the quality by performing data wrangling.
- I then started exploring the processed data. I used SQL to query the data and gather insights.
- I gained further insights into the data by applying statistical analysis and data visualization, to see how variables might be related to each other.
- I finally applied machine learning algorithms to make predictive analysis.

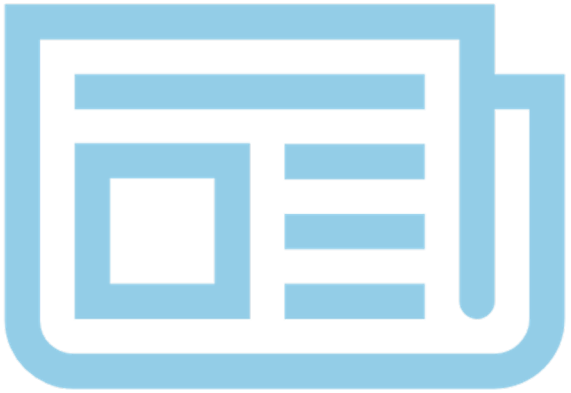
Introduction

In this Project, I will predict if the Falcon 9 first stage will land successfully.

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.

Methodology



- Data collection methodology:
  - Data was collected using the open-source SpaceX REST API
  - Webscraping a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches
- Perform data wrangling
  - Json_normalize was used to get the data into a flat table
  - Deal with null values
  - Sampling the data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Create a column for the class
  - Standardize the data
  - Split into training data and test data
  - Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
  - Find the method performs best using test data

Data collection

Data was collected using the open-source SpaceX REST API

Webscraping a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches also could provide the same data

I performed a get request using the requests library to obtain the launch data, which I used to get the data from the API.

This result can be viewed by calling the .json() method.

The response was in the form of a JSON, specifically a list of JSON objects;

- To convert this JSON to a dataframe, I used the json_normalize function.
- This function allowed me to “normalize” the structured json data into a flat table.

Data collection – SpaceX API

Notebook:

https://github.com/nbatobs/IBM-Watson/blob/master/Capstone/Data%20Collection.ipynb

```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

```
# 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'])
```



```
# 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']:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

```
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

Data collection – Web scraping

https://github.com/nbatobs/IBM-Watson/blob/master/Capstone/Web%20Scraping.ipynb

```
def date_time(table_cells):
    """
    This function returns the data and time from the HTML  table cell
    Input: the  element of a table data cell extracts extra row
    """
    return [data_time.strip() for data_time in list(table_cells.strings)][0:2]
```

```
def booster_version(table_cells):
    """
    This function returns the booster version from the HTML  table cell
    Input: the  element of a table data cell extracts extra row
    """
    out=''.join([booster_version for i,booster_version in enumerate( table_cells.strings) if i%2==0][0:-1])
    return out
```

```
def landing_status(table_cells):
    """
    This function returns the landing status from the HTML table cell
    Input: the  element of a table data cell extracts extra row
    """
    out=[i for i in table_cells.strings][0]
    return out
```

```
def get_mass(table_cells):
    mass=unicodedata.normalize("NFKD", table_cells.text).strip()
    if mass:
        mass.find("kg")
        new_mass=mass[0:mass.find("kg")+2]
    else:
        new_mass=0
    return new_mass
```

```
def extract_column_from_header(row):
    """
    This function returns the landing status from the HTML table cell
    Input: the  element of a table data cell extracts extra row
    """
    if (row.br):
        row.br.extract()
    if row.a:
        row.a.extract()
    if row.sup:
        row.sup.extract()

    colunm_name = ' '.join(row.contents)

    # Filter the digit and empty names
    if not(colunm_name.strip().isdigit()):
        colunm_name = colunm_name.strip()
        return colunm_name
```

NOTE: Explanations are in red

Data wrangling

- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident.
- I converted those outcomes into training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

- You need to present your data wrangling process using key phrases and flowcharts

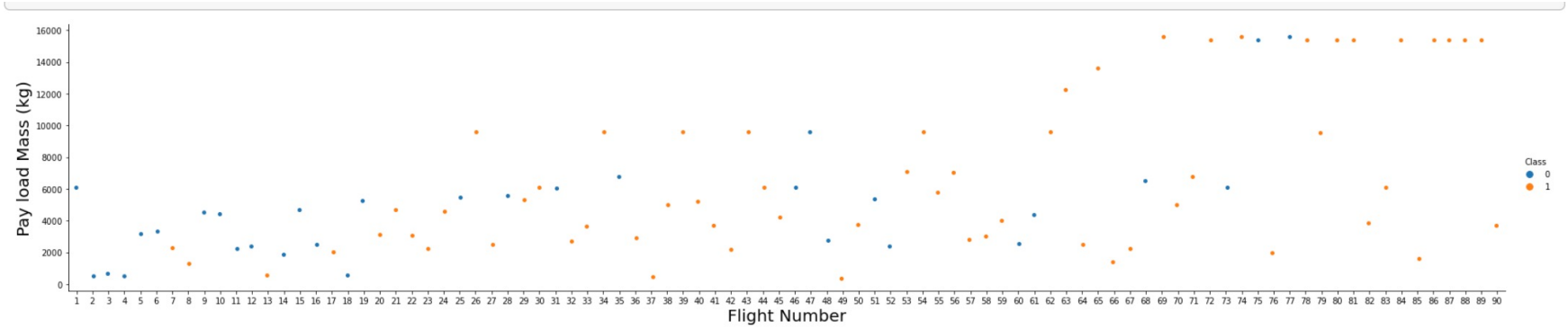
- https://github.com/nbatobs/IBM-Watson/blob/master/Capstone/Data%20Wrangling.ipynb

EDA with data visualization

I plotted the FlightNumber vs. PayloadMass and overlayed the outcome of the launch.

We see that as the flight number increases, the first stage is more likely to land successfully.

The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

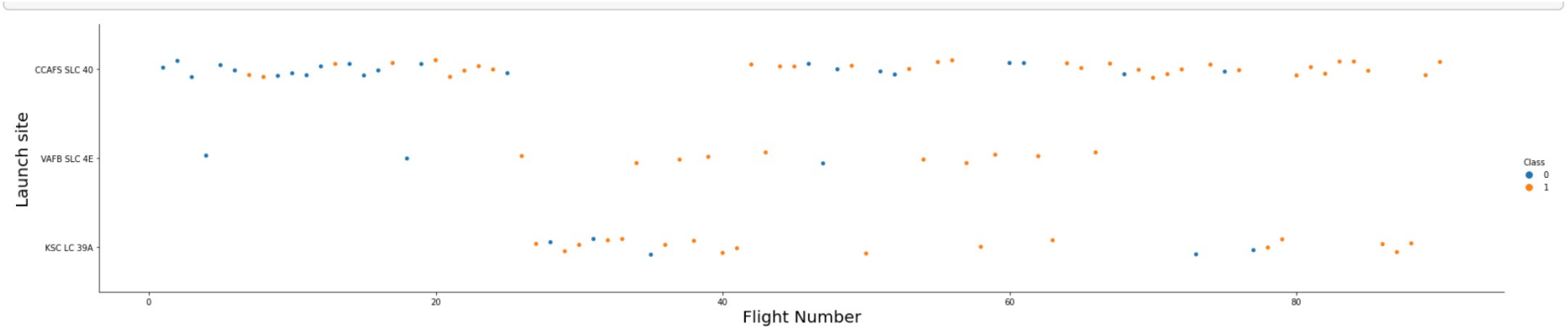


Notebook Link:

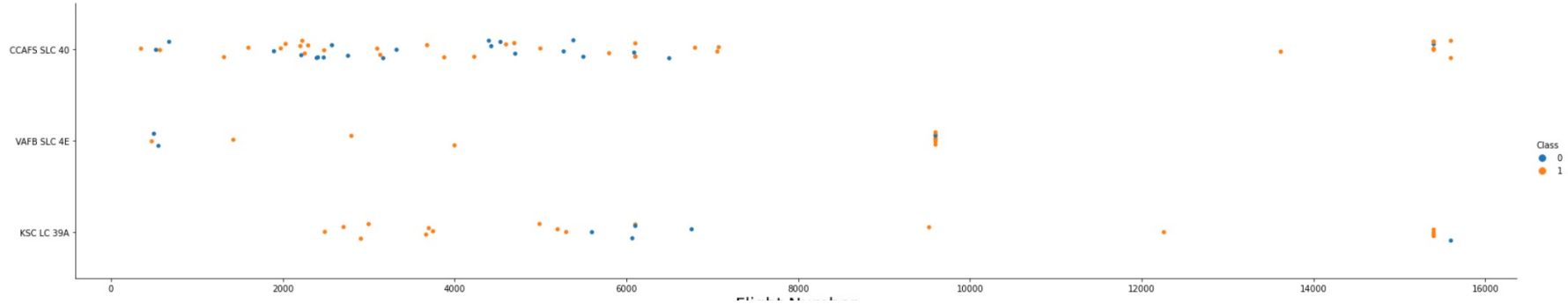
https://github.com/nbatobs/IBM-Watson/blob/master/Capstone/EDA%20with%20Visualisation.ipynb

EDA with data visualization

For the relationship between Flight Number and Launch Site we also see that as the flight number increases, the first stage is more likely to land successfully.

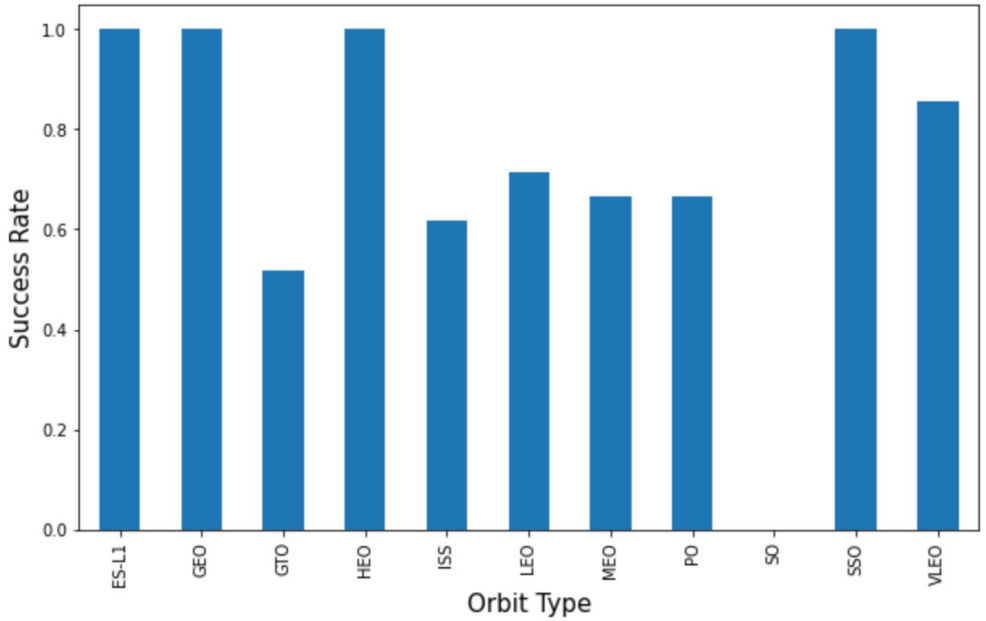


For the relationship between Payload and Launch Site we see that a Higher payload is associated with a higher success rate

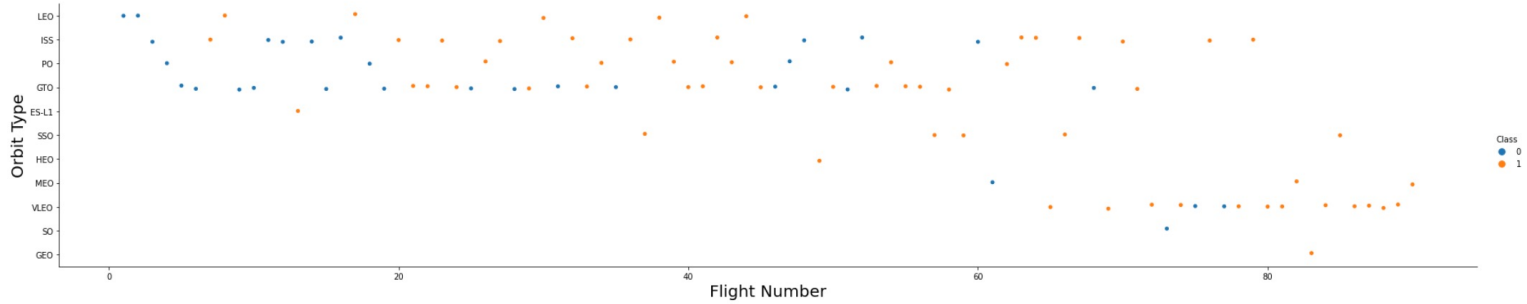


EDA with data visualization

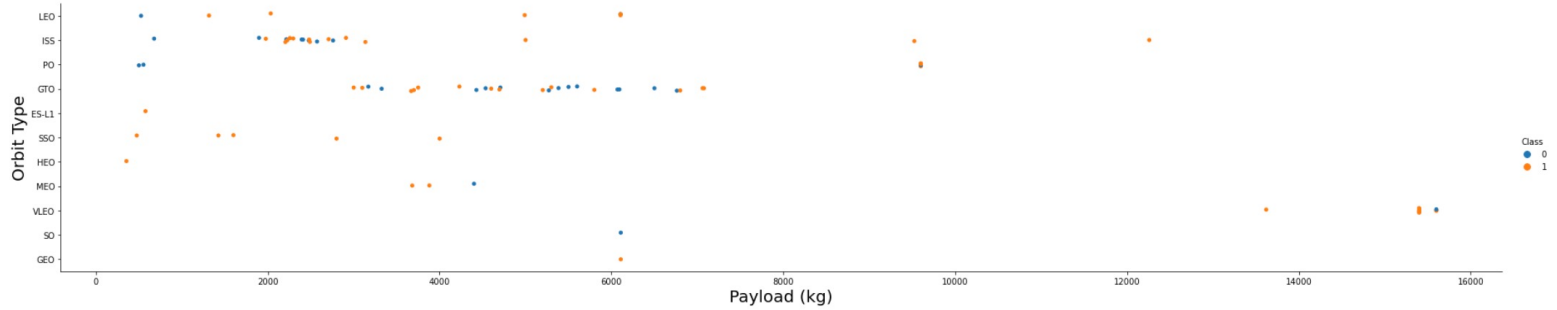
For the relationship between success rate of each orbit type we see which orbits have the best success rates



The LEO orbit success rate appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.



EDA with SQL

- *Display the names of the unique launch sites in the space mission*
- *Display 5 records where launch sites begin with the string 'CCA'*
- *Display the total payload mass carried by boosters launched by NASA (CRS)*
- *Display average payload mass carried by booster version F9 v1.1*
- *List the date when the first successful landing outcome in ground pad was achieved.*
- *List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000*
- *List the total number of successful and failure mission outcomes*
- *List the names of the booster versions which have carried the maximum payload mass. Use a subquery*
- *List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015*
- *Rank the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.*

Note: All the performed queries are available in the notebook below

https://github.com/nbatobs/IBM-Watson/blob/master/Capstone/EDA%20with%20SQL.ipynb

Predictive analysis (Classification)

Steps

Perform exploratory Data Analysis and determine Training Labels

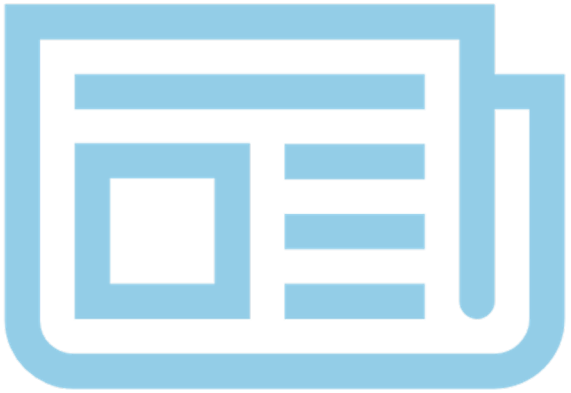
- Created a column for the class
- Standardized the data
- Split into training data and test data

Found the best Hyperparameter for SVM, Classification Trees and Logistic Regression

Find the method performs best using test data.

❖ **Please find the link to the notebook below:**
https://github.com/nbatobs/IBM-Watson/blob/master/Capstone/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

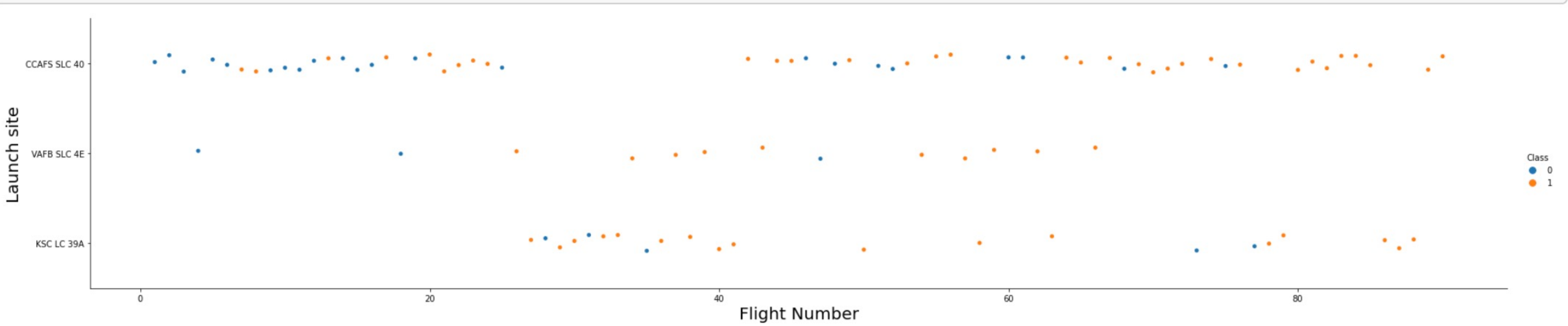
Results



- Exploratory data analysis results
- Predictive analysis results

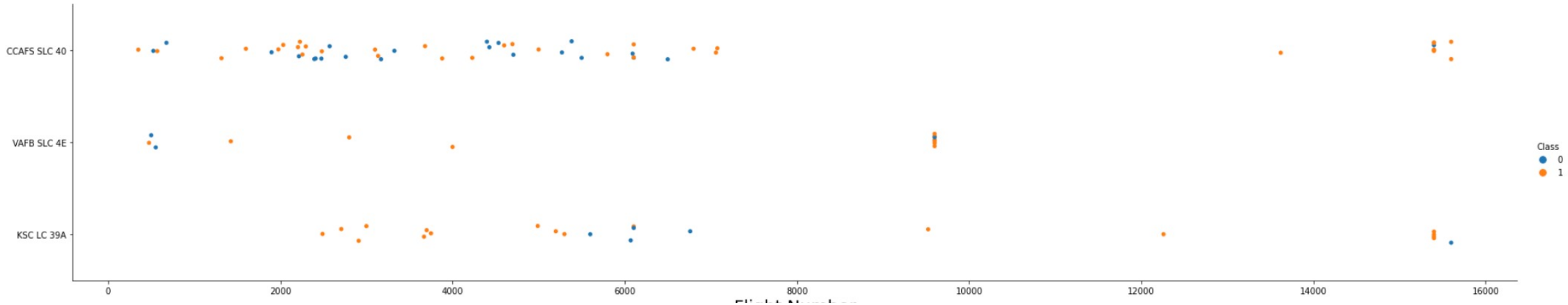
EDA with Visualization

Flight Number vs. Launch Site



For the relationship between Flight Number and Launch Site we also see that as the flight number increases, the first stage is more likely to land successfully.

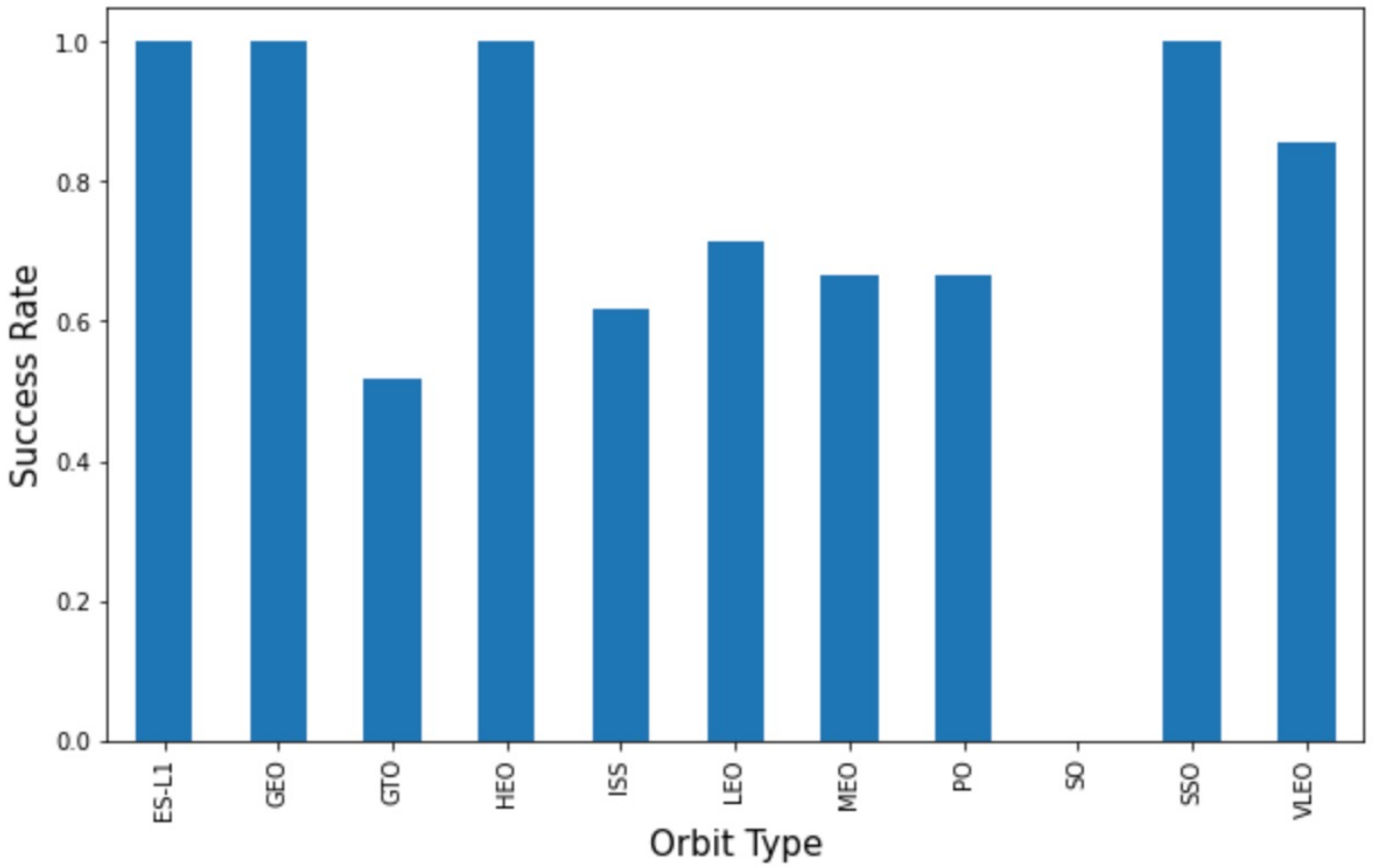
Payload vs. Launch Site



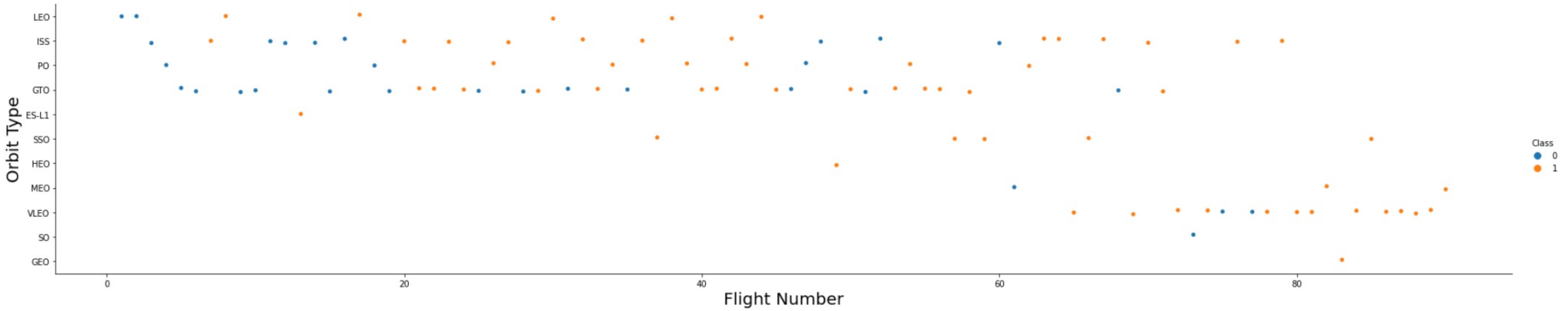
For the relationship between Payload and Launch Site we see that a Higher payload is associated with a higher success rate

Success rate vs. Orbit type

For the relationship between success rate of each orbit type we see which orbits have the best success rates

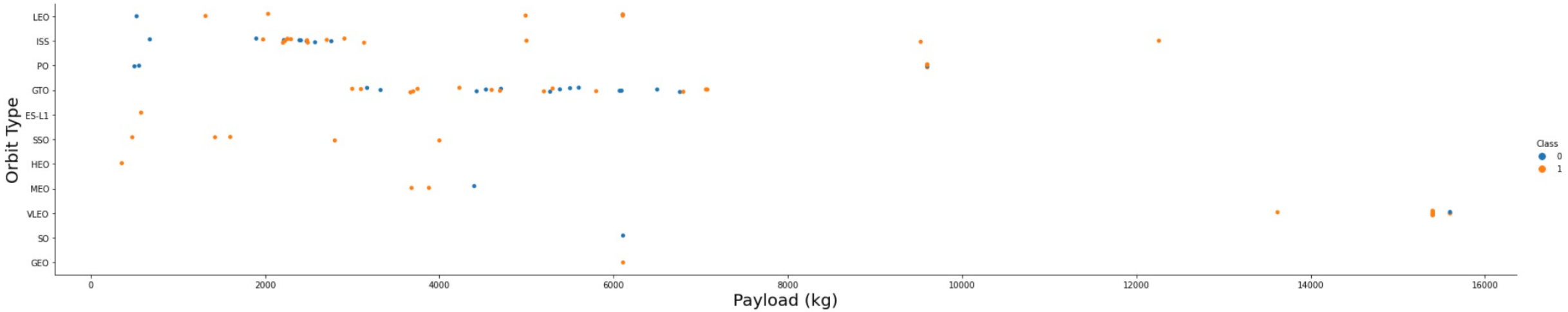


Flight Number vs. Orbit type



The LEO orbit success rate appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit type



Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

EDA with SQL

All launch site names

```
%sql SELECT DISTINCT launch_site FROM SPACEXTBL
```

```
 * ibm_db_sa://vtl93937:***@0c77d6f2-5da9-48a9-81f8-86b520b87518
Done.
```

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

- Query to find the names of the unique launch sites

Launch site names begin with `CCA`

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

```
 * ibm_db_sa://vtl93937:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

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

- Query to find all launch sites begin with `CCA`

Total payload mass

```
%sql SELECT SUM(payload_mass_kg) AS mass FROM SPACEXTBL WHERE customer LIKE '%(CRS)%'
```

```
 * ibm_db_sa://vtl93937:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.d
Done.
```

mass
48213

- *Query to calculate the total payload mass carried by boosters launched by NASA (CRS)*

Average payload mass by F9 v1.1

```
%sql SELECT AVG(payload_mass_kg) AS mass_of_booster FROM SPACEXTBL WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://vtl93937:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdo
Done.
```

mass_of_booster
2928

- Query to calculate the average payload mass carried by booster version F9 v1.1

First successful ground landing date

```
%sql SELECT MIN(DATE) AS DATE FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success%'
```

```
 * ibm_db_sa://vtl93937:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.
Done.
```

DATE
2015-12-22

- Query to find the date when the first successful landing outcome in ground pad

Successful drone ship landing with payload between 4000 and 6000

```
%sql SELECT booster_version FROM SPACEXTBL WHERE landing_outcome = 'Success (drone ship)' AND payload_mass_kg >4000 <6000
```

```
 * ibm_db_sa://vtl93937:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

booster_version
F9 FT B1021.1
F9 FT B1023.1
F9 FT B1029.2
F9 FT B1038.1
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

- Query to list the names of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

Total number of successful and failure mission outcomes

```
%sql SELECT mission_outcome, COUNT mission_outcome FROM SPACEXTBL GROUP BY mission_outcome ORDER BY mission_outcome
```

```
 * ibm_db_sa://vtl93937:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

mission_outcome	mission_outcome_1
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- Query to calculate the total number of successful and failure mission outcomes

Boosters carried maximum payload

```
%sql SELECT booster_version, payload_mass_kg  FROM SPACEXTBL WHERE payload_mass_kg =(SELECT MAX(payload_mass_kg) FROM SPACEXTBL)
```

```
 * ibm_db_sa://vtl93937:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

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

- Query to list the names of the boosters which have carried the maximum payload mass

2015 launch records

```
%%sql SELECT MONTHNAME(DATE) AS Month, landing_outcome, booster_version, launch_site FROM SPACEXTBL
WHERE landing_outcome ='Failure (drone ship)'
```

```
 * ibm_db_sa://vtl93937:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.
Done.
```

MONTH	landing_outcome	booster_version	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
January	Failure (drone ship)	F9 v1.1 B1017	VAFB SLC-4E
March	Failure (drone ship)	F9 FT B1020	CCAFS LC-40
June	Failure (drone ship)	F9 FT B1024	CCAFS LC-40

- Query to list the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015

Rank success count between 2010-06-04 and 2017-03-20

```
%%sql SELECT landing_outcome, COUNT landing_outcome FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing_outcome
HAVING landing_outcome LIKE '%Success%'
```

```
 * ibm_db_sa://vtl93937:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2ic
Done.
```

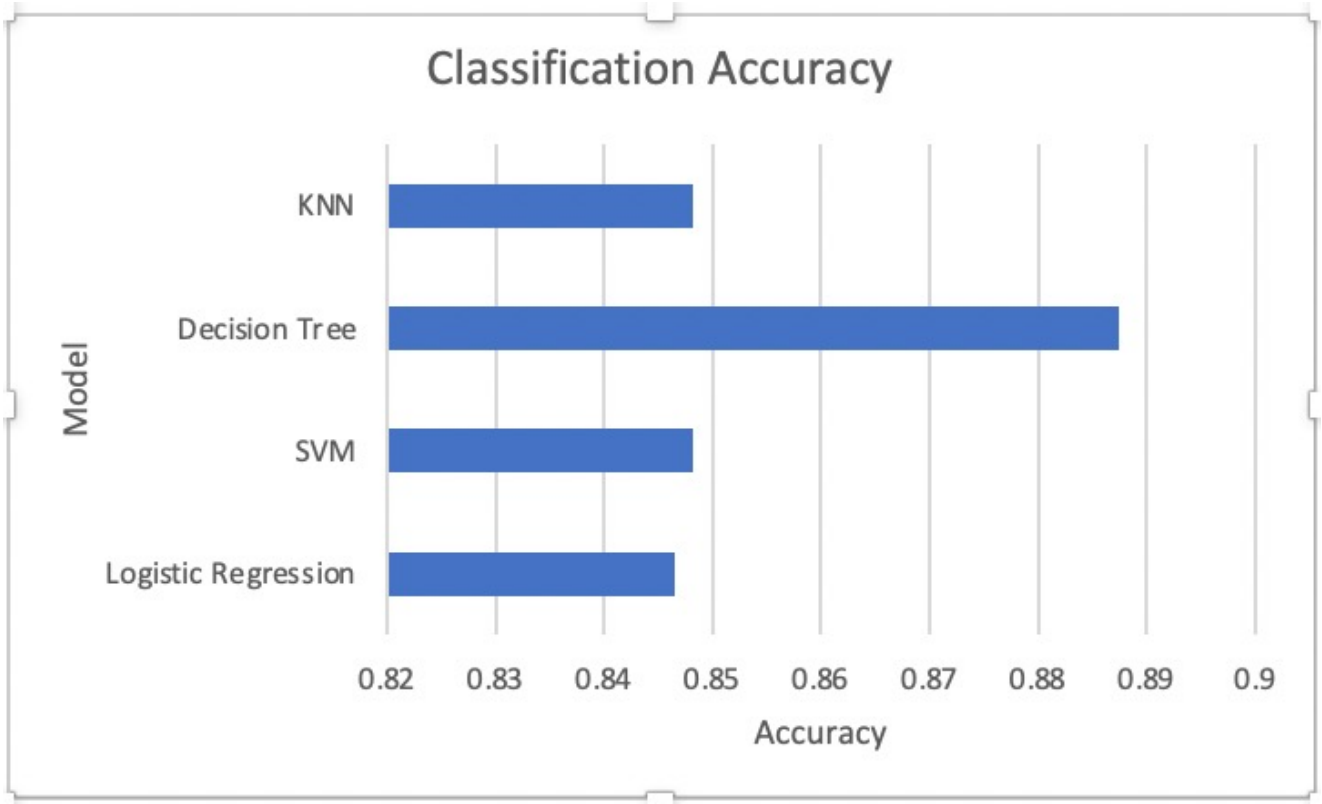
landing_outcome	landing_outcome_1
Success (drone ship)	5
Success (ground pad)	3

- Query to rank the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

Predictive analysis (Classification)

Classification Accuracy

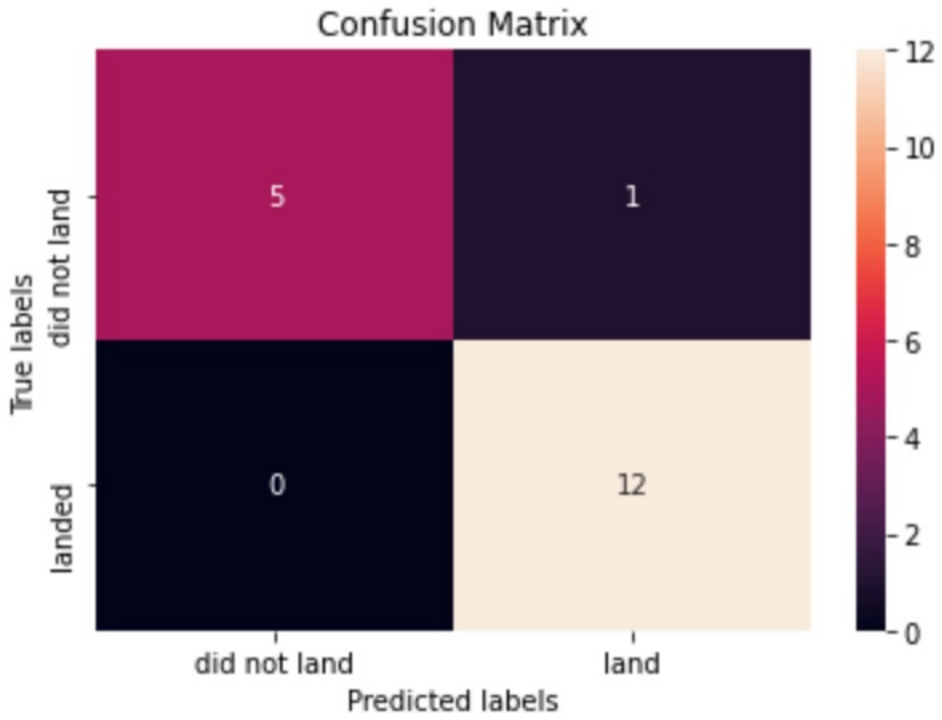
The Decision Tree Classification method had the highest model accuracy



Confusion Matrix

On the right is the confusion matrix of the KNN model, which was the highest accuracy on the test data.

- It predicted the first stage to land 13 times where the actual value was 12
- It predicted the first stage to not land 5 times where the actual value was 6.



CONCLUSION

In this Project, I attempted to predict if the Falcon 9 first stage will land successfully.
If we can determine if the first stage will land, we can determine the cost of a launch.

- Obtaining the data through an API was easier for me than webscraping.
- I found, using data visualization, the important variables to be pay load mass, orbit, and launch site.
- The KNN model showed to have the best accuracy on the test data with a test accuracy of 88%
- This model can help us determine if the first stage will land, thus we can determine the cost of a launch.

APPENDIX

- The file that contains all notebooks is:

https://github.com/nbatobs/IBM-Watson/tree/master/Capstone

- I would like to thank my family for their assistance throughout the course