IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hubert Fernando Pérez Pérez
04/01/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with Data Visualization
  - Exploratory Data Analysis with SQL
  - Interactive Visual Analytics with Folium
  - Predictive Analysis with Machine Learning
- Summary of all results
  - Exploratory Data Analysis
  - Interactive Analytics in Screenshots
  - Predictive Analytics Results

Introduction

- Project background and context

  Space X announces on its website Falcon 9 rocket launches at a cost of $62 million, with other providers the cost is over $165 million each, much of the savings is due to Space X being able to reuse the first stage of the rocket. Therefore, if we can determine if it will land the first stage, we can determine the cost of a launch. This information can be used if an alternative company wants to bid against Space X for a rocket launch. The goal of the project is to create a machine learning study to predict whether the first stage will land successfully.

- Problems you want to find answers
  - The interaction between various characteristics that determine the success rate of a successful landing.
  - What operational conditions must be met to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

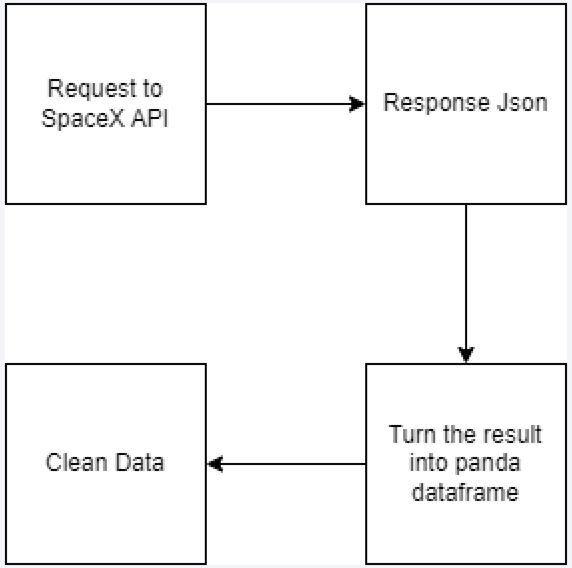
- Data collection methodology:
  - The data was collected using the SpaceX API and Wikipedia web scraping.
- Perform data wrangling
  - One-hot coding was applied to categorical characteristics.
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
  - The data collection was performed via GET request to the SpaceX API.
  - We then decoded the contents of the response as a Json using the .json() function call and activatedin a pandas data frame using .json_normalize().
  - We then clean up the data, check for missing values and fill in missing values.where necessary.
  - We also performed web scraping of Wikipedia to obtain Falcon 9 launch records.
  - The goal was to extract the launch records as an HTML table, parse the table and convert it into a pandas data frame for future analysis.

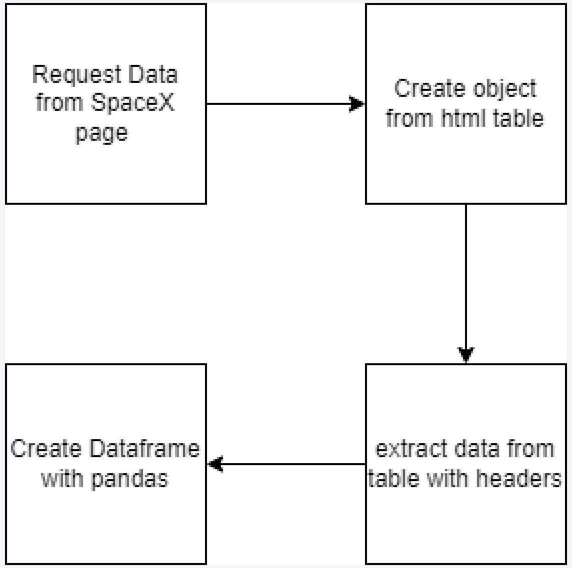
Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
    - We used the GET request to the SpaceX API to collect data, cleaned the requested data and did some data arrays and formatting, finally we stored it in a data frame.
- https://github.com/Hiufer/CourseraFinalProject/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb



Data Collection - Scraping

- We applied web scraping to the Falcon 9 launch records.
- We analyzed the table and converted into a pandas data frame.
- https://github.com/Hiufer/CourseraFinalProject/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb



Data Wrangling

- Describe how data were processed
  - We performed exploratory data analysis and determined the formation of thelabels.
  - We calculate the number of launches at each site, and the number and occurrence of each orbit.
  - We create a target result label from the result column and export it in csv.

- https://github.com/Hiufer/CourseraFinalProject/blob/main/Exploratory%20Data%20Analysis%20for%20Data%20Visualization.ipynb

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
  - We explored the data by visualizing the relationship between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, and annual trend of launch success.

- https://github.com/Hiufer/CourseraFinalProject/blob/main/Exploratory%20Data%20Analysis%20for%20Data%20Visualization.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed

- We loaded the SpaceX dataset into a SQL database without leaving the Jupyter notebook.
- We applied EDA with SQL to get information from the data. We write queries to find out, for example:
    - The name of the unique launch sites on the space mission.- The total mass of payload carried by NASA launched boosters (CRS).
- https://github.com/Hiufer/CourseraFinalProject/blob/main/Exploratory%20Data%20Analysis%20for%20Data%20Visualization.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- We mark all launch sites and add map objects such as markers, circles and lines to mark the success or failure of launches for each site on the folium map.
- We assign the function launch results (failure or success) to classes 0 and 1.
- Using color-labeled groups of markers, we identify which launch sites have relatively high success rates.

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Building an interactive dashboard with Plotly Dash
- We plotted pie charts showing the total releases for given sites.
- We plotted a scatter plot showing the relationship to output and payload.

- https://github.com/Hiufer/CourseraFinalProject/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tuned different hyperparameters.
- We used accuracy as a metric for our model, improved the model by feature engineering and algorithm tuning.

- https://github.com/Hiufer/CourseraFinalProject/blob/main/Machine%20Learning%20Prediction.ipynb

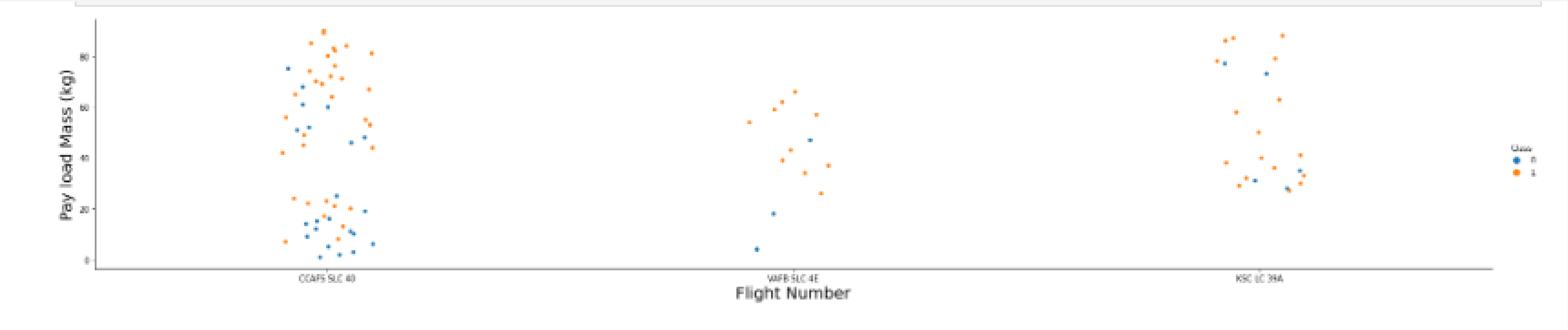
Results

- Exploratory data analysis results
  - Interactive analytical data demonstration
  - Data results with predictive analytics

Section 2

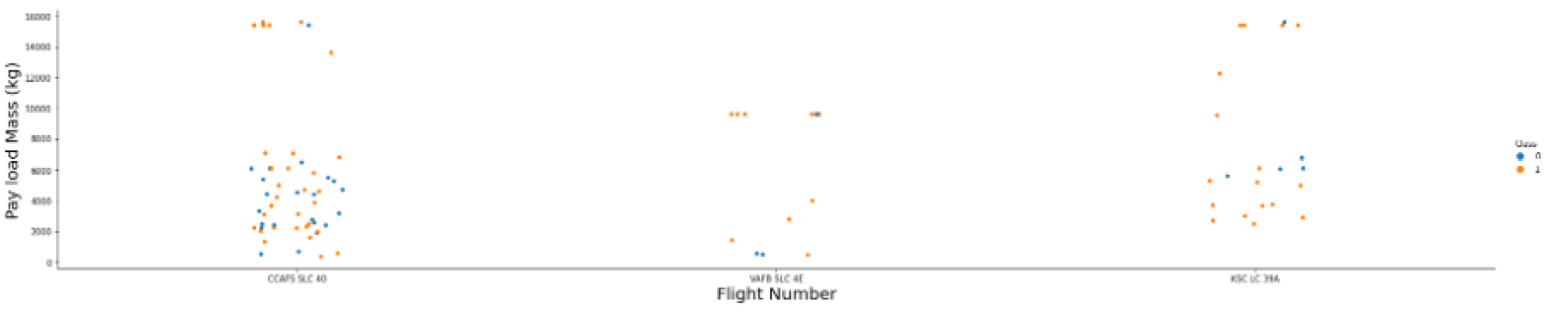
Insights drawn from EDA

Flight Number vs. Launch Site



We found that the higher the number of flights at a launch site, the higher the success rate at a launch site.

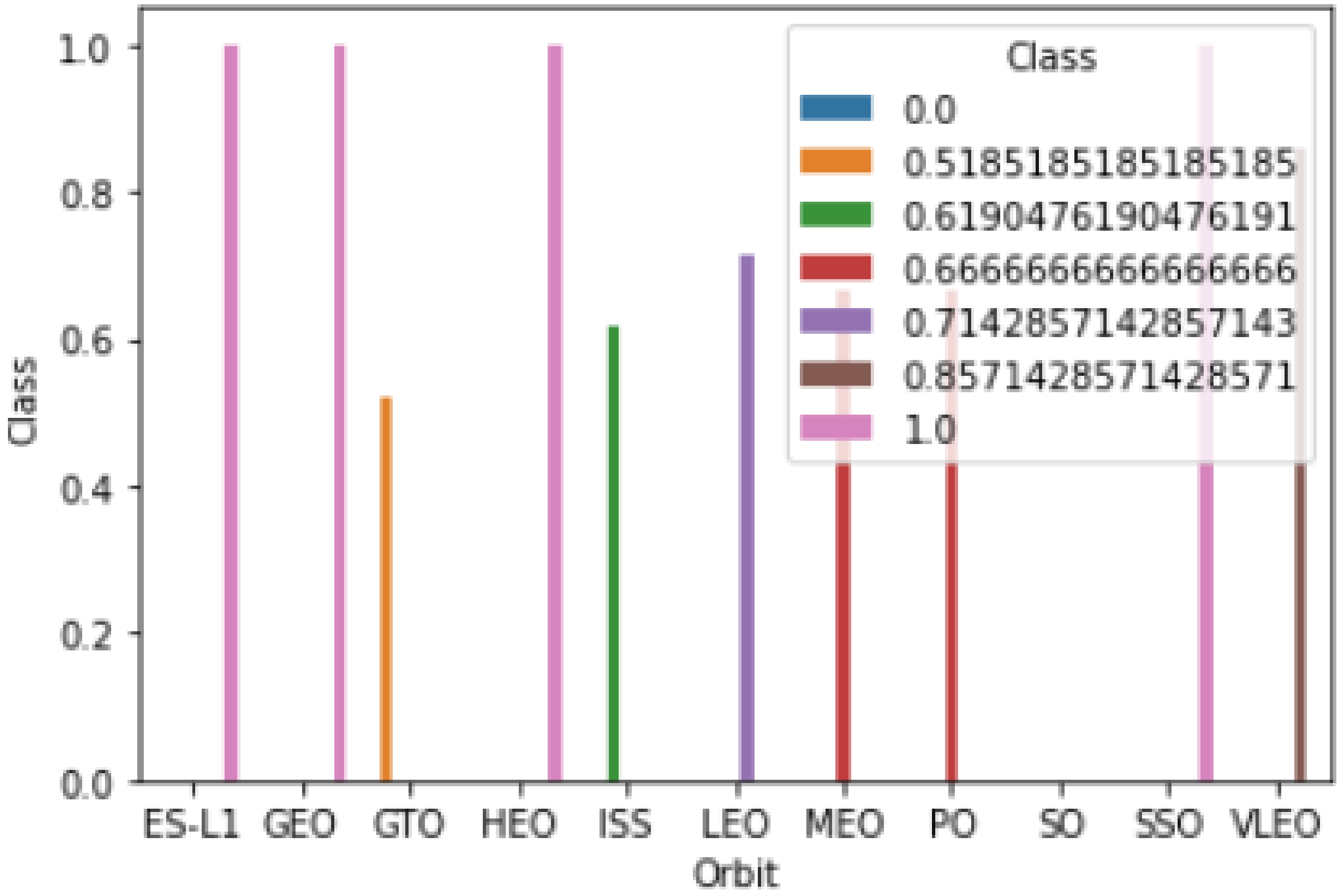
Payload vs. Launch Site

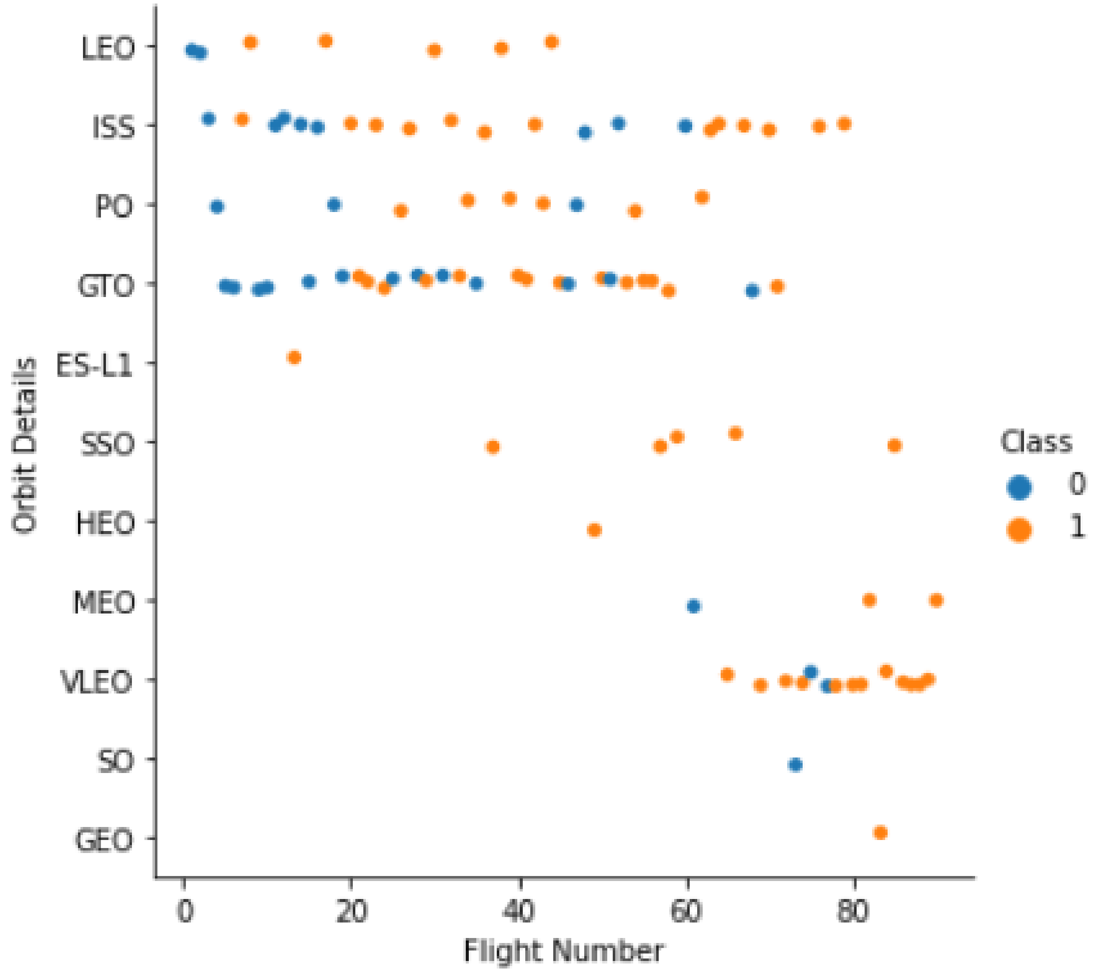


It was found that the higher the payload mass for the CCAFS SLC 40 launch site, the higher the rocket success rate.

Success Rate vs. Orbit Type

we can see that ES-L1, GEO, HEO, SSO, VLEO had the highest rate of success



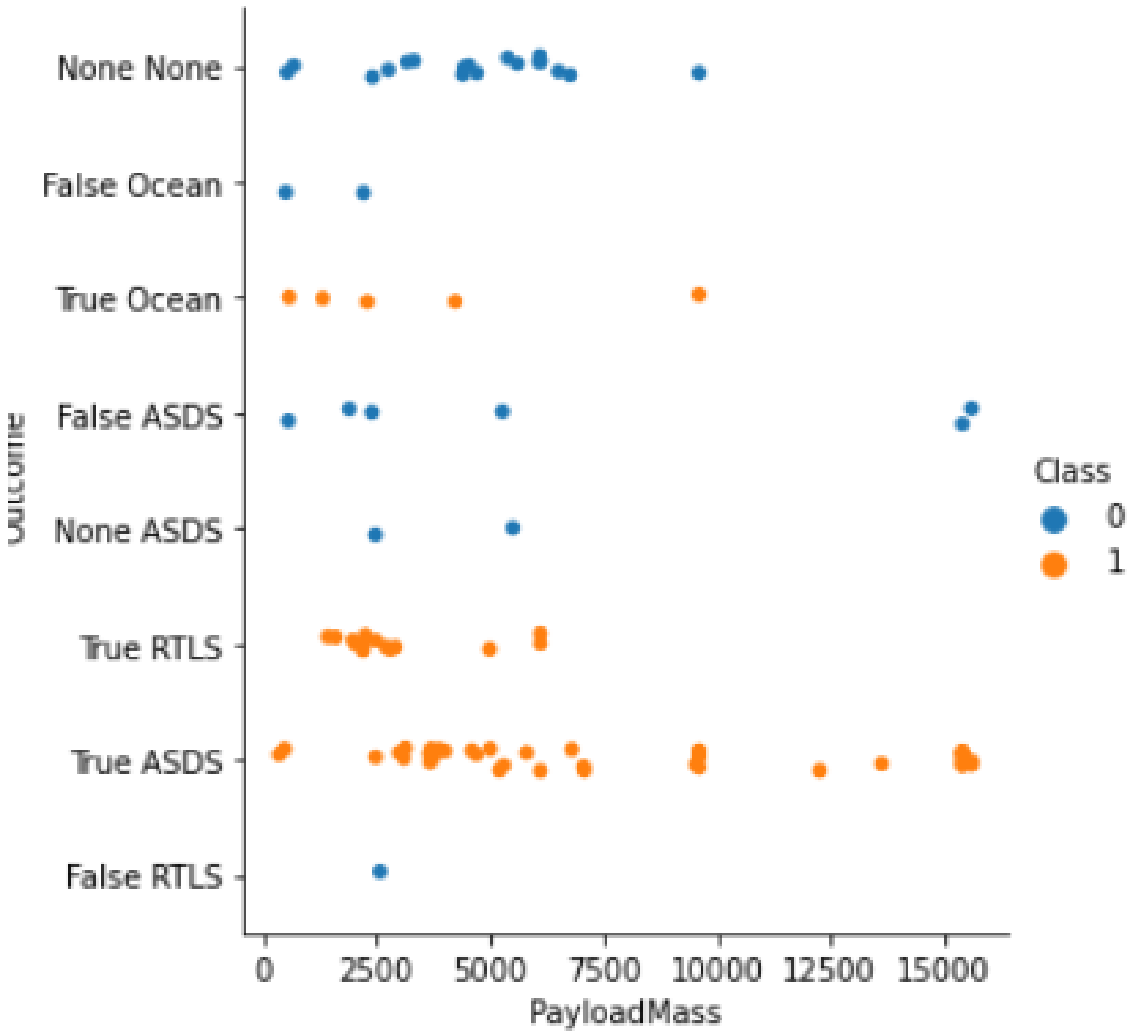


Flight Number vs. Orbit Type

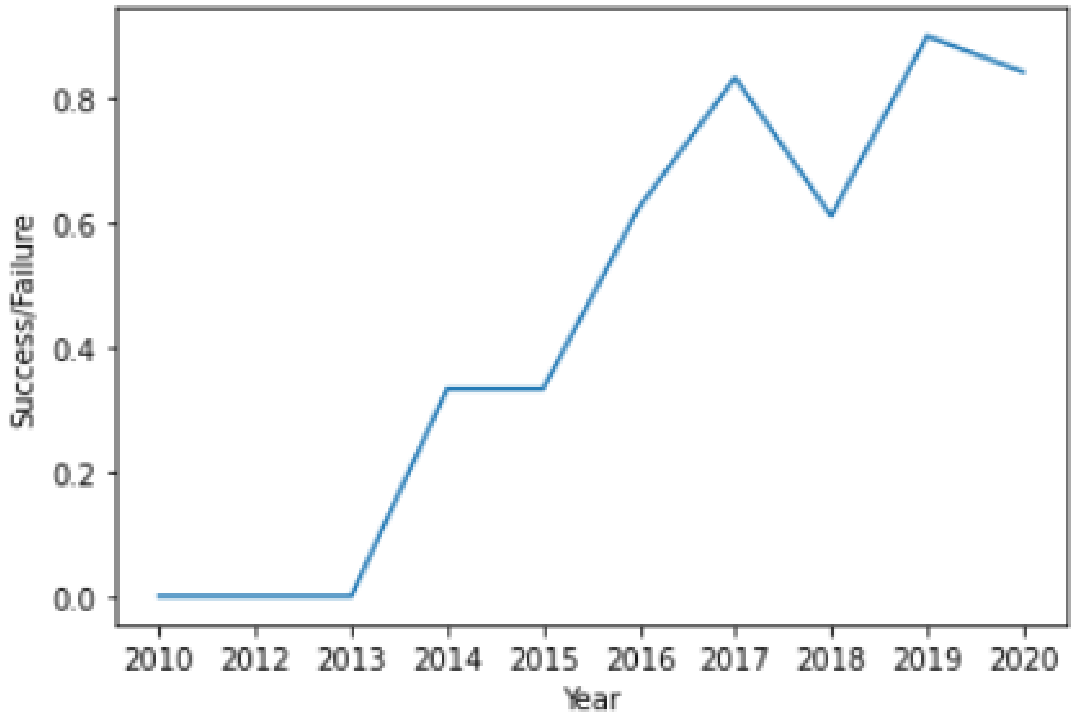
We note that in LEO orbit, success is related to the number of flights, while in GTO orbit there is no relationship between the number of flights and the orbit.

Payload vs. Orbit Type

In the graph we can observe that with heavy payloads, successful landings are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend



We can observe that the success rate since 2013 continued to increase until 2020.

All Launch Site Names

Display the names of the unique launch sites in the space mission

```
%%sql
select distinct LAUNCH_SITE
from SPACEXTBL;
```

```
   ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
 * ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/blud
b;security=SSL
Done.
```

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

Launch Site Names Begin with 'CCA'

the table shows the 5 records where the launch sites begin with 'CCA'.

Out[31]:

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

Total Payload Mass

The total for NASA propellant is 45596 using the following query

```
%%sql
select sum(PAYLOAD_MASS__KG_)
from SPACEXTBL
where Customer = 'NASA (CRS)';
```

```
 ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.c
loud:31505/bludb
 * ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.c
loud:31505/bludb;security=SSL
Done.
```

1
45596

Average Payload Mass by F9 v1.1

The average payload mass transported by the F9 v1.1 reinforcement version is 2534

```
%%sql
select AVG(payload_mass__kg_) as avg from SPACEXTBL
where booster_version like 'F9 v1.1%'
```

```
   ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
 * ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb;security=SSL
Done.
```

AVG
2534

First Successful Ground Landing Date

DISTINCT was used to find the correct value representing a successful landing on the ground and then we used the MIN function to find the dates of the first successful landing result on the ground platform.

```
%%sql
select distinct landing__outcome from SPACEXTBL
```

```
  ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appc
loud:31505/bludb
 * ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appc
loud:31505/bludb;security=SSL
Done.
```

landing__outcome
Controlled (ocean)
Failure
Failure (drone ship)
Failure (parachute)
No attempt
Precluded (drone ship)
Success
Success (drone ship)
Success (ground pad)
Uncontrolled (ocean)

```
%%sql
select min(date) from SPACEXTBL where landing__outcome = 'Success (ground pad)'
```

```
  ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appc
loud:31505/bludb
 * ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appc
loud:31505/bludb;security=SSL
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

The WHERE clause was used to filter thrusters that successfully landed on an unmanned ship and the AND condition was applied to determine successful landing with a payload mass between 4000 and 6000.

```
%%sql
select booster_version, payload_mass__kg_ from SPACEXTBL
where landing__outcome = 'Success (drone ship)' and 4000 < payload_mass__kg_ and payload_mass__kg_ < 6000
group by booster_version, payload_mass__kg_
```

```
 ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
* ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb;security=SSL
Done.
```

booster_version	payload_mass__kg_
F9 FT B1021.2	5300
F9 FT B1031.2	5200
F9 FT B1022	4696
F9 FT B1026	4600

Total Number of Successful and Failure Mission Outcomes

Grouped by mission_outcome to see which missions have failed and which have succeeded

```
%%sql
select mission_outcome, count(mission_outcome) as total_nr
from SPACEXTBL
group by mission_outcome
```

```
* ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.c
loud:31505/bludb
Done.
```

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

Boosters Carried Maximum Payload

We determine the booster that has carried the maximum payload by using a subquery in the WHERE clause and the MAX() function to obtain the maximum payload_mass__kg_.

```
%%sql
SELECT DISTINCT booster_version
FROM SPACEXTBL
WHERE payload_mass__kg_ = (
    SELECT max(payload_mass__kg_)
    FROM SPACEXTBL
)
```

```
 * ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

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

2015 Launch Records

A combination of WHERE, AND and BETWEEN clause was used to filter results of failed landings on unmanned spacecraft for 2015.

```
%%sql
select landing__outcome, booster_version,launch_site
from SPACEXTBL
where landing__outcome = 'Failure (drone ship)' and year(date) = 2015
group by landing__outcome, booster_version,launch_site
```

```
 * ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

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

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

GROUP BY and ORDER BY were used to sort the count of landing results as Failure or Success between June 4, 2016 and March 20, 2010 in descending order.

```
%%sql
select landing__outcome, count(landing__outcome) as total_nr
from SPACEXTBL
where date between '2010-06-04' and '2017-03-20'
group by landing__outcome
order by total_nr desc
```

```
 * ibm_db_sa://jxd70927:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

landing__outcome	total_nr
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

Section 3

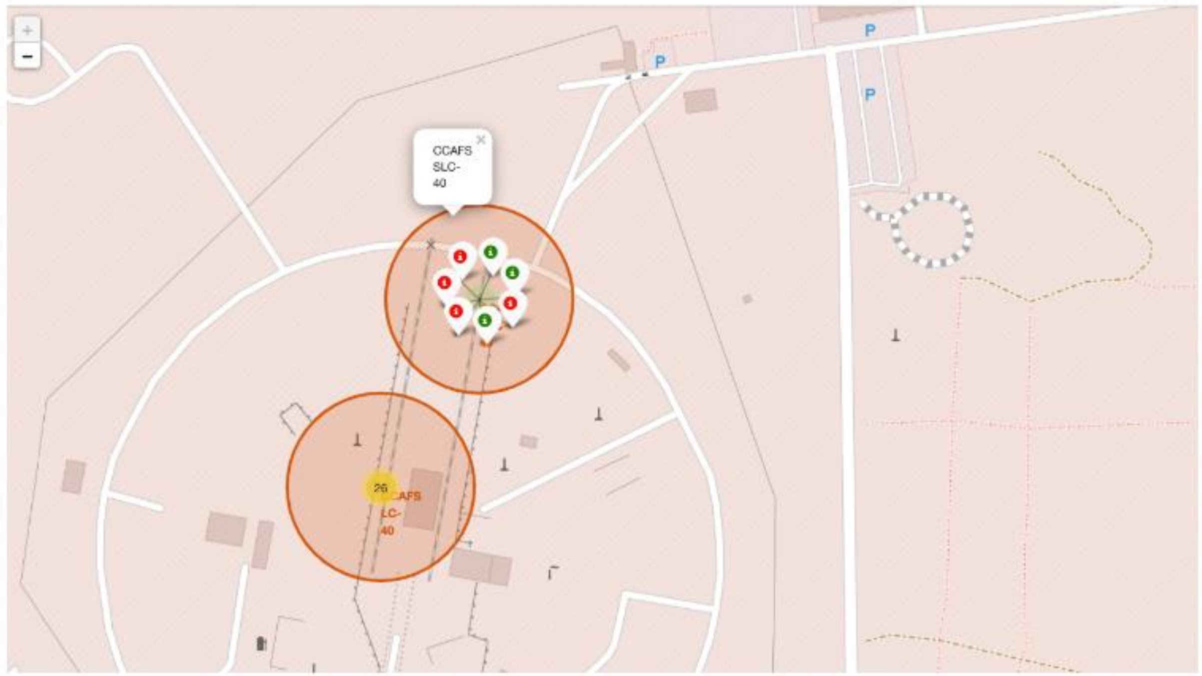
Launch Sites Proximities Analysis

SpaceX Launch

Graph of all SpaceX launch sites are on the U.S. coasts, Florida and California.

Color-labeled launches

Successful launches are marked in green and unsuccessful launches in red.



<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

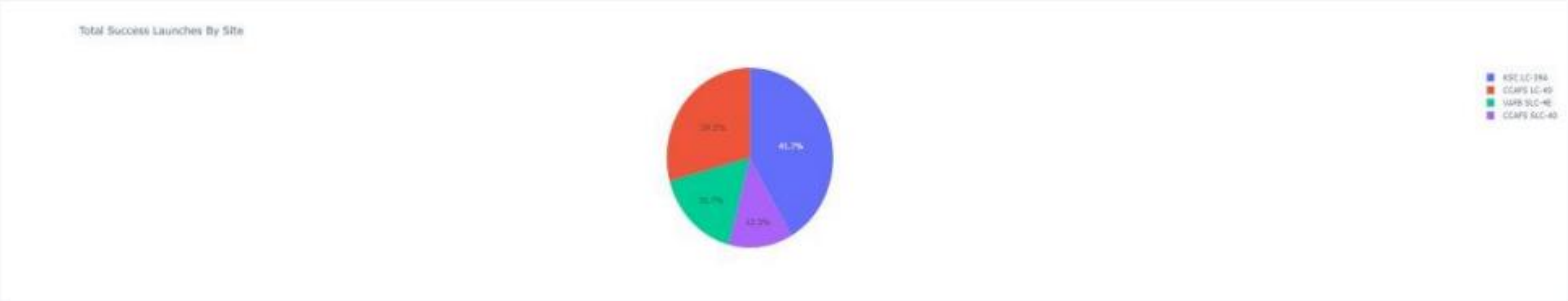
- Explain the important elements and findings on the screenshot

Section 4

Build a Dashboard with Plotly Dash

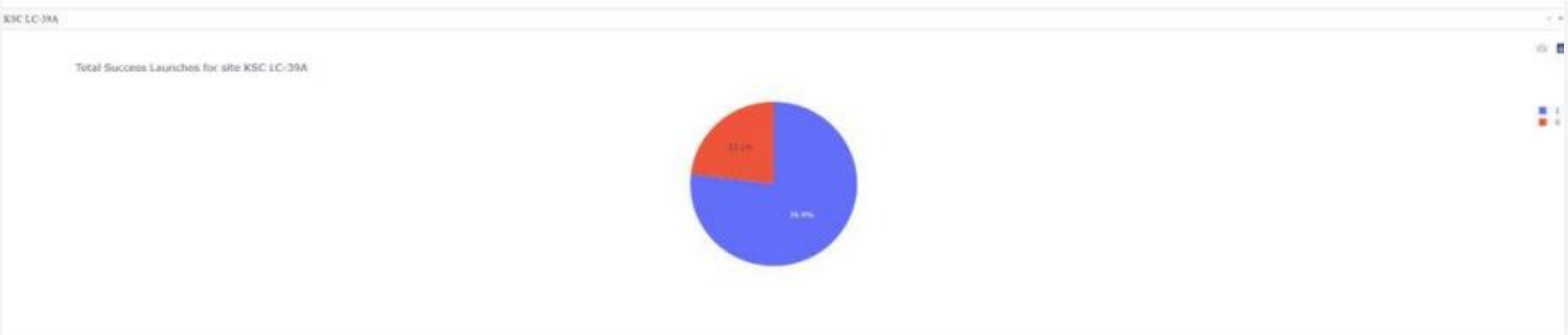
Total Success

It was classified in which places have been the most launches.



KSC Launch

It was filtered what the percentage of output was at the KSC LC-39A site



Section 5

Predictive Analysis (Classification)

Classification Accuracy

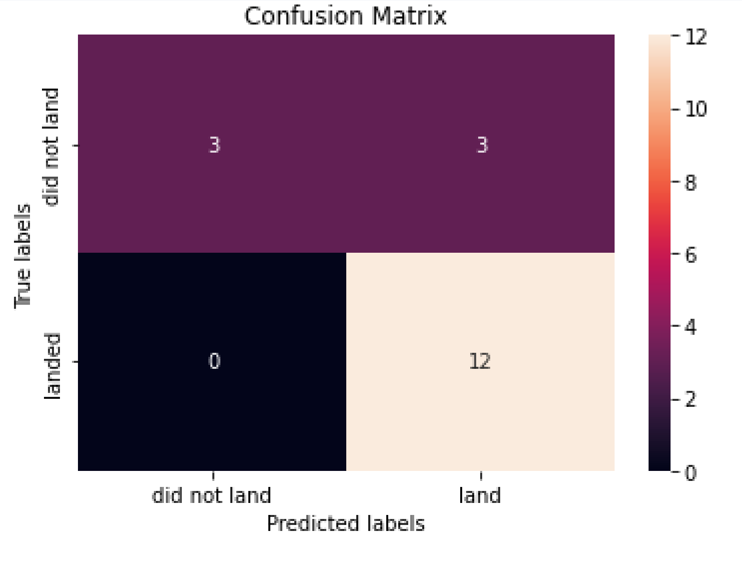
The decision tree classifier is the model with the highest classification accuracy.

```python
# After comparing accuracy of above methods, they all preformed practically
# the same, except for tree which fit train data slightly better but test data worse.
models = {'LogisticRegression':logreg_cv.best_score_,
          'SupportVectorMachine': svm_cv.best_score_,
          'DecisionTree':tree_cv.best_score_,
          'KNeighbours':knn_cv.best_score_
         }
bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVectorMachine':
    print('Best params is :', svm_cv.best_params_)
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbours':
    print('Best params is :', knn_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8767857142857143
Best params is : {'criterion': 'gini', 'max_depth': 2, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_sa
mples_split': 10, 'splitter': 'best'}
```

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation
- The confusion matrix of the decision tree classifier shows that the classifier can distinguish between the different classes. The main problem is false positives



Conclusions

The conclusions drawn from my work are as follows:

- The higher the number of flights at a launch site, the higher the success rate at a launchsite.
- The launch success rate started to increase from 2013 to 2020.
- The decision tree classifier is the best machine learning algorithm for this task.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

- https://github.com/Hiufer/CourseraFinalProject

Thank you!