Data Science Capstone project

Denis Ovchinnikov

August 9, 2021

Outline

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

Executive Summary

- We use python and jupyter notebooks to gather, process and clean the data about Falcon 9 launches.
- We use SQL, Folium and Dash to explore and visualize the data.
- We evaluate several prediction models to predict whether a future launch will be successful.

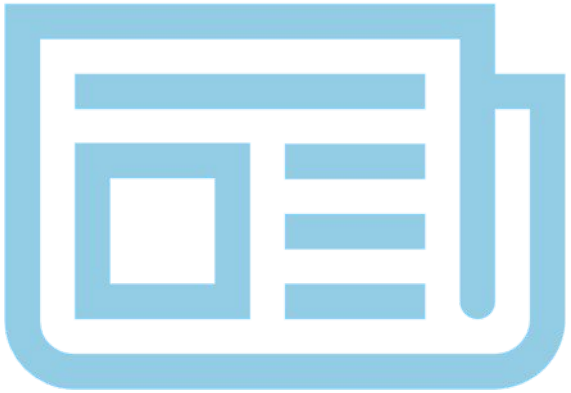
- We choose KNN model with 10 neghboors as the model. It performs with the accuracy of 83% on the test set.

Introduction

- SpaceX redefined what is an adequate cost for rocket launches. While a launch costs over $165,000,000 with other providers, SpaceX advertises to have reduced this to $62,000,000. Much of the savings comes from the ability to reuse the first stage of the rocket.

- In this project we use publicly available data to analyze successful retrievals of the first stage after Falcon 9 launches. We also develop predictive models to predict whether a retrieval of the first Stage would be successful in the future, given the information about the launch.

Methodology



- Data collection methodology:
  - SpaceX API data was collected as JSON using requests package
  - SpaceX Wikipedia page data was collected using web scrapping
- Data wrangling:
  - We analyzed the outcome feature of each launch and determined what constitutes a successful retrieval of the first stage. We then created a new Boolean column that has 1 if the retrieval was successful and 0 otherwise
  - We studied how different features impact the outcome and selected those features that have a reasonable impact as training labels
  - We converted categorical features to numerical using one-hot-encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We separated all data into train and test sets, and built four models on the train set
  - Fitting was done by using 10-fold cross-validation

Methodology

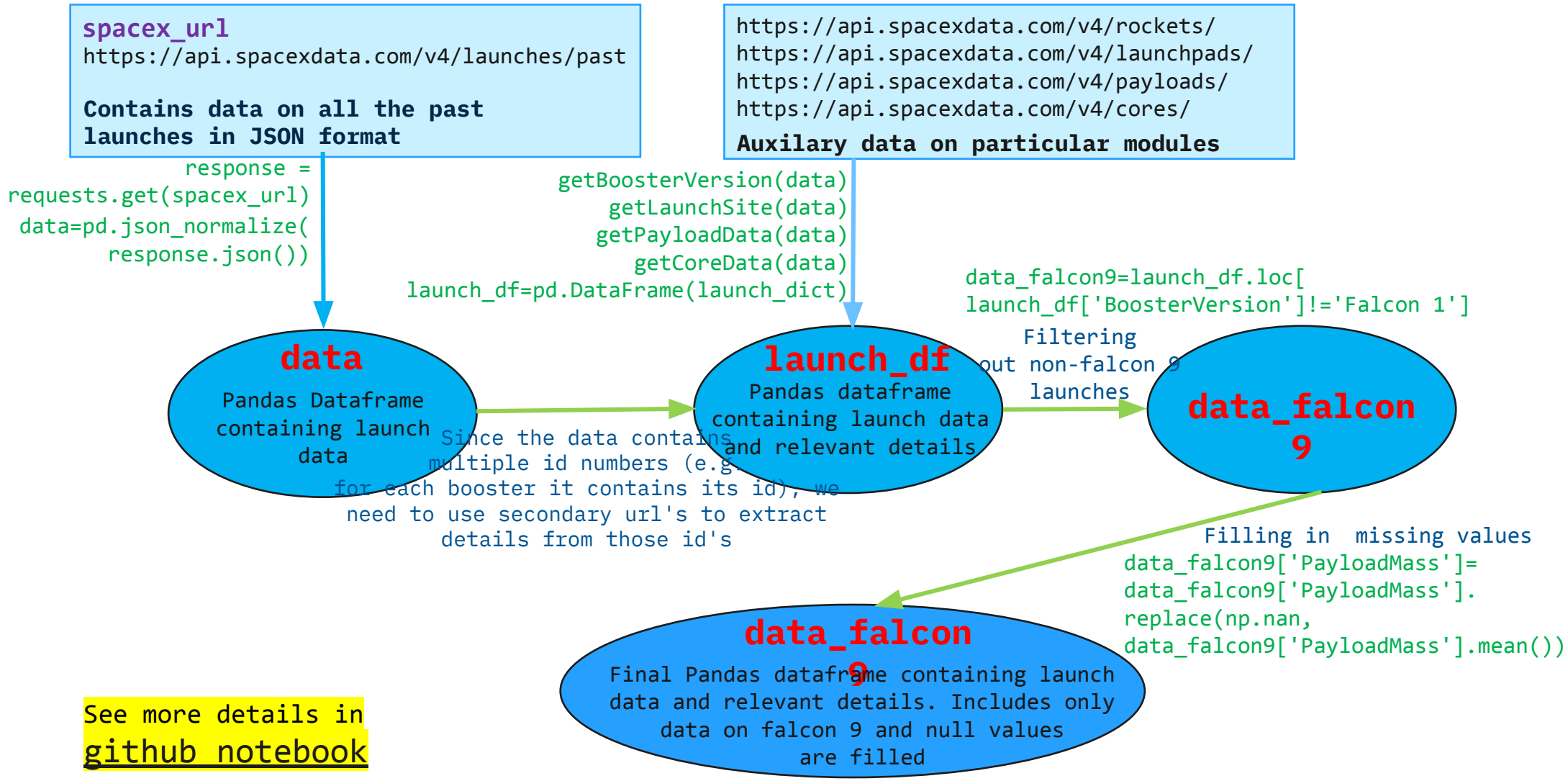
Here we briefly discuss the main methodology used throughout the project.

Data collection

We worked with two datasets:

- SpaceX API data was collected as JSON using requests package
- SpaceX Wikipedia page data was collected using web scraping

Data collection – SpaceX API



See more details in github notebook

Data collection – Web scraping

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

Static version of wikipedia page that contains data on all the past launches in JSON format

```
raw_data=requests.get(static_url)
soup_data=BeautifulSoup(raw_data.content)
html_tables=soup_data.find_all('table')
column_names = []
for header in first_launch_table.find_all('th'):
    col_name=extract_column_from_header(header)
    if col_name!=None:
        column_names.append(col_name)
```

column_names
Names of all the columns we want to fetch

For each row in each of the tables, extract
Corresponding columns in the list in the launch_dict, e.g:

```
for rows in table.find_all("tr"):
payload = row[3].a.string
 launch_dict['Payload'].append(payload)
```

launch_dict
Dictionary with column names as keys and empty list values

launch_dict
Updated dictionary filled with data from the tables

Filtering out non-falcon 9 launches

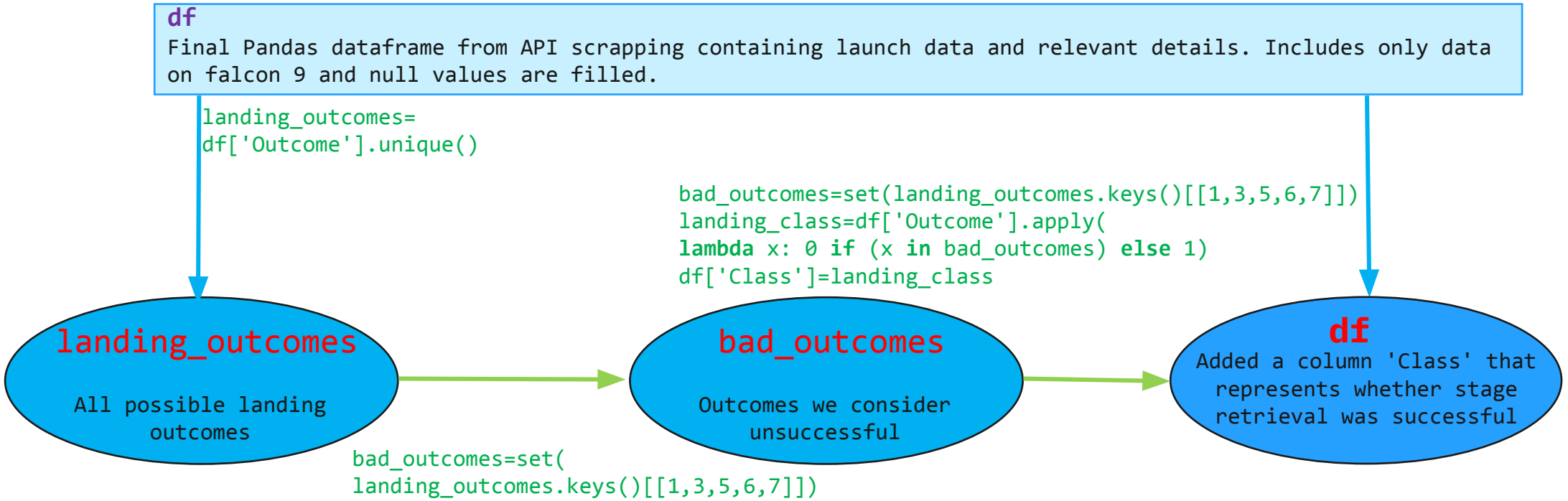
Converting to DataFrame

```
df=pd.DataFrame(
launch_dict)
```

df
Final Pandas dataframe containing launch data and relevant details.

See more details in github notebook

Data wrangling



See more details in github notebook

10

EDA with data visualization

We plotted several charts to figure out what features impact the success of retrieving the rocket (column 'Class').

Most of the charts were scatter plots with each data point colored according to the Class feature.

- Flight Number vs Payload Mass scatter graph
- Flight Number vs Launch Site scatter graph
- Payload Mass vs Launch Site scatter graph
- Flight Number vs Orbit type scatter graph
- Payload Mass vs Orbit type scatter graph
- Success rate vs Orbit type bar chart
- Year vs Success rate line graph

- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

See more details in
github notebook

EDA with SQL

We performed several SQL queries to learn insights about data:

- Names of the unique launch sites in the space mission
- 5 records where launch sites begin with the string 'CCA'
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1
- Date when the first succesful landing outcome in ground pad was achieved
- Names of the 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
- Names of the booster_versions which have carried the maximum payload mass
- Month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
- Count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 ranked in descending order

See more details in github notebook

An interactive map with Folium

- We created an interactive map displaying the following objects:
  - Each of the launch sites present in the database
  - For each of the sites, a marker cluster that contains a marker for each of the launches from that site. The marker is colored according to the success of the corresponding mission
  - For one of the sites (KSC LC-39A), we added a line from that site to the closest vital infrastructure objects. Each of the lines displays an icon with the distance to the object, and a mouse-over tooltip explaining what does the line represent
- The map helps us visualize the dataset and inspect some interesting features about sites locations, e.g. the fact that three of the sites are located within 10 km of each other.

See more details in github notebook or in ibm notebook (contains interactive maps)

Dashboard with Plotly Dash

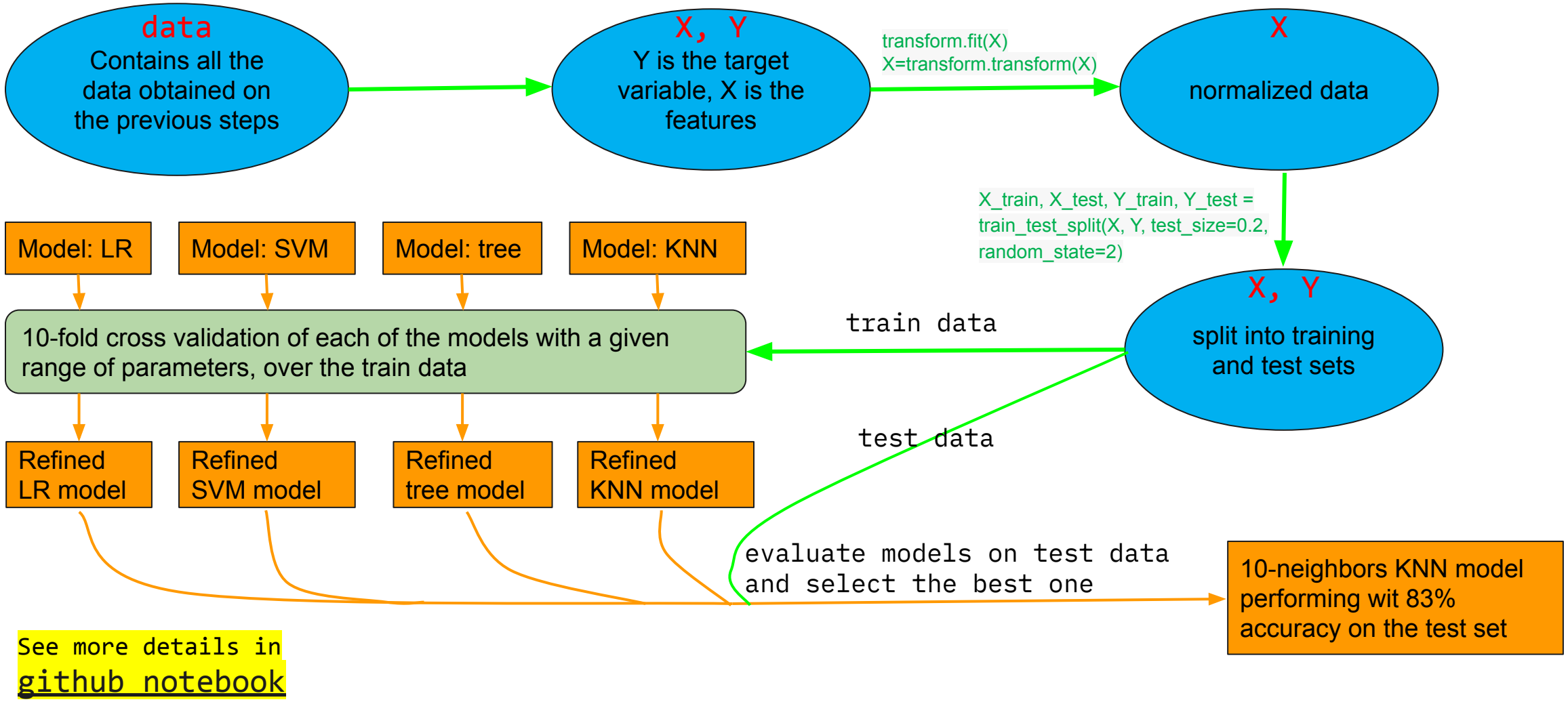
The dashboard includes the following functionalities:

- Dropdown for the site selection (either a particular site or all sites)
- A Pie chart that displays:
  a) The number of successful launches per site if all sites option is selected
  b) Successful vs unsuccessful launches from a specific site if that site is selected
- Payload mass range slider
- A scatter-plot of Payload Mass vs Class each point colored according to the booster version used. If a specific site or range of payload mass is selected, the graph dispays only that data that falls within chosen bounds.

The dashboard allows us to study how do the launch site, payload mass and the booster version affect outcome of the launches. For example, we see that site KSC LC-39A has the most successful launches, as well as the best success rate (76.9%).

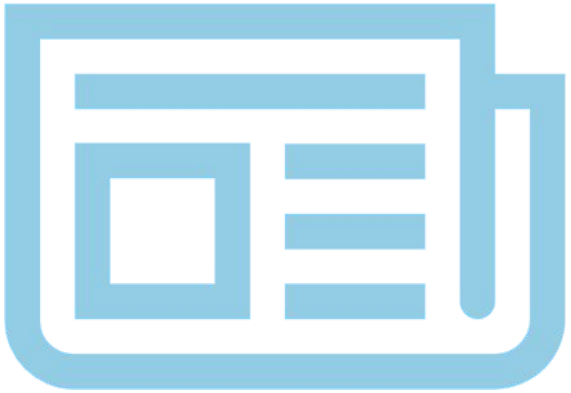
See more details in
github python script

Predictive analysis (Classification)



See more details in github notebook

Results



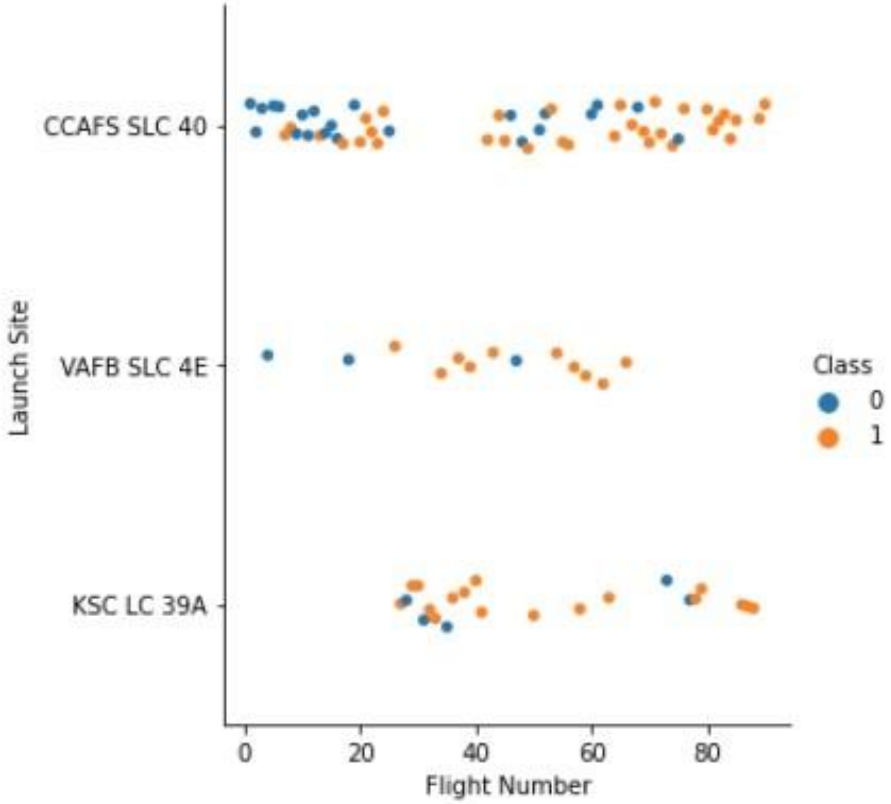
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

EDA with Visualization

We use a jupyter notebook to visualize the dataset. The data is stored as a Pandas DataFrame in a variable df.

Flight Number vs. Launch Site

```
sns.catplot(data=df,x='FlightNumber', y='LaunchSite', hue='Class')
plt.xlabel('Flight Number')
plt.ylabel('Launch Site')
plt.show()
```



We see that initially most launches were performed from CCAFS SLC 40 site, but most of the flights between number 20 and number 40 were performed from KSC LC 39A site.

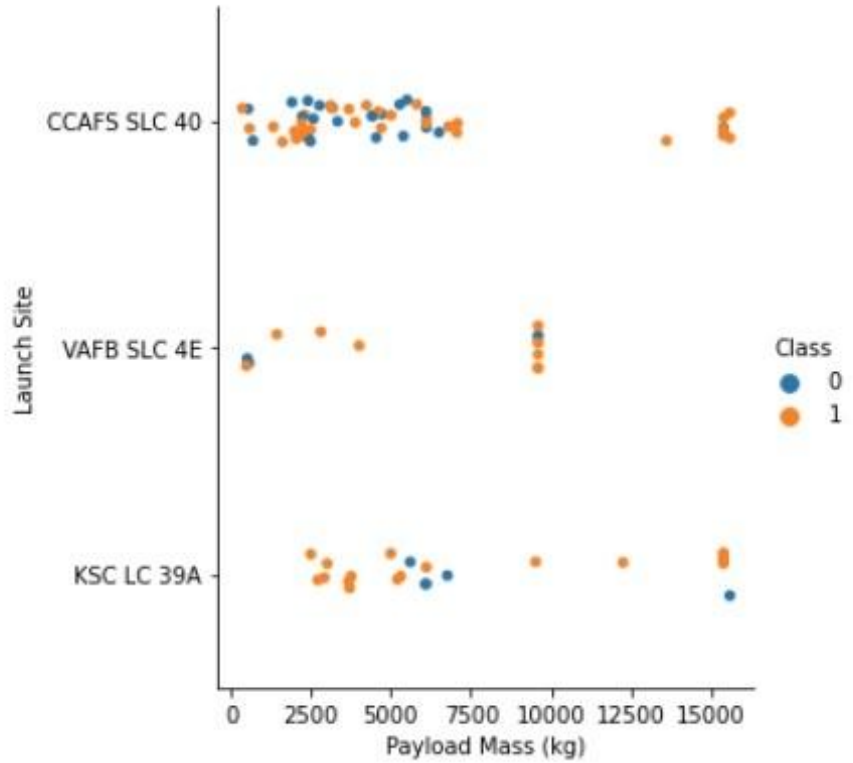
We also see that visually, the success rate of launches from KSC LC 39A is the best.

Payload vs. Launch Site

We see that the heaviest launches happened from CCAFS and KSC, but VAFB facilitated majority of mid-weight launches.

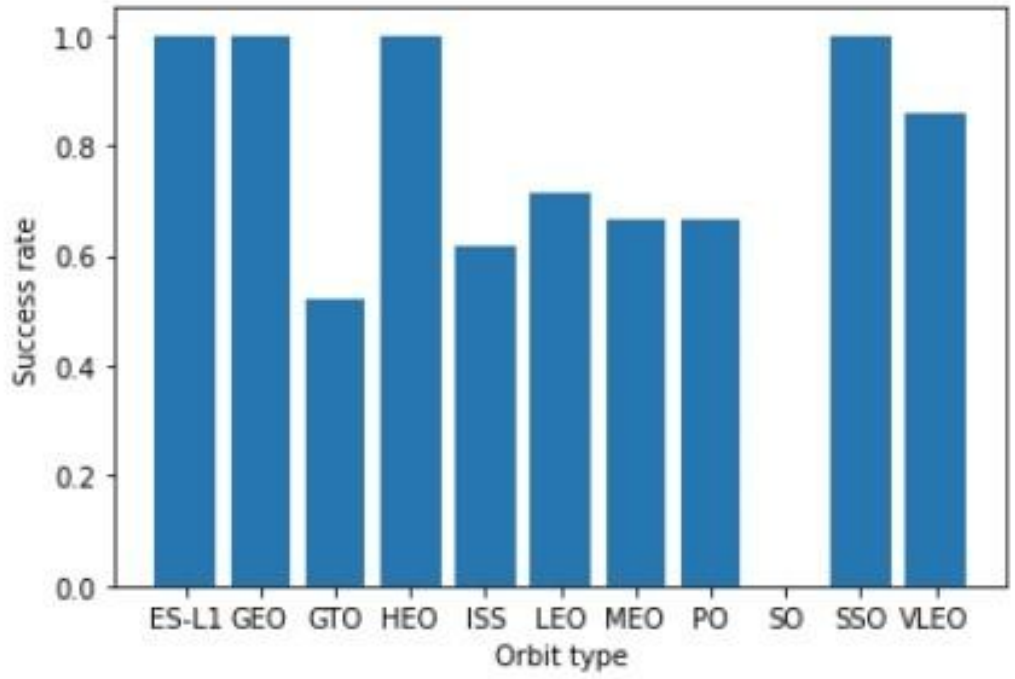
In fact, since the payload masses for those launches from VAFB SLC 40 align perfectly (exactly 9000 kg), it was likely due to a series of repetitive missions. Inspecting Wikipedia data confirms that those were 5 launches for "Iridium NEXT" project (famous satellites that are often seen with a naked eye).

```
sns.catplot(data=df,x='PayloadMass', y='LaunchSite', hue='Class')
plt.xlabel('Payload Mass (kg)')
plt.ylabel('Launch Site')
plt.show()
```



Success rate vs. Orbit type

```
df_suc_per_orbit=df.groupby('Orbit')['Class'].mean().reset_index()
plt.bar(df_suc_per_orbit['Orbit'], df_suc_per_orbit['Class'])
plt.xlabel('Orbit type')
plt.ylabel('Success rate')
plt.show()
```



We see that several orbits have perfect success rate of 100% (ES-L1, GEO, HEO, SSO). The orbit with the lowest non-zero success rate is GTO. We also see that SO orbit has 0% success rate.

Next slide illustrates that most of those animalic success rates(100% and 0%) happen for orbits that has only one launch corresponding to them

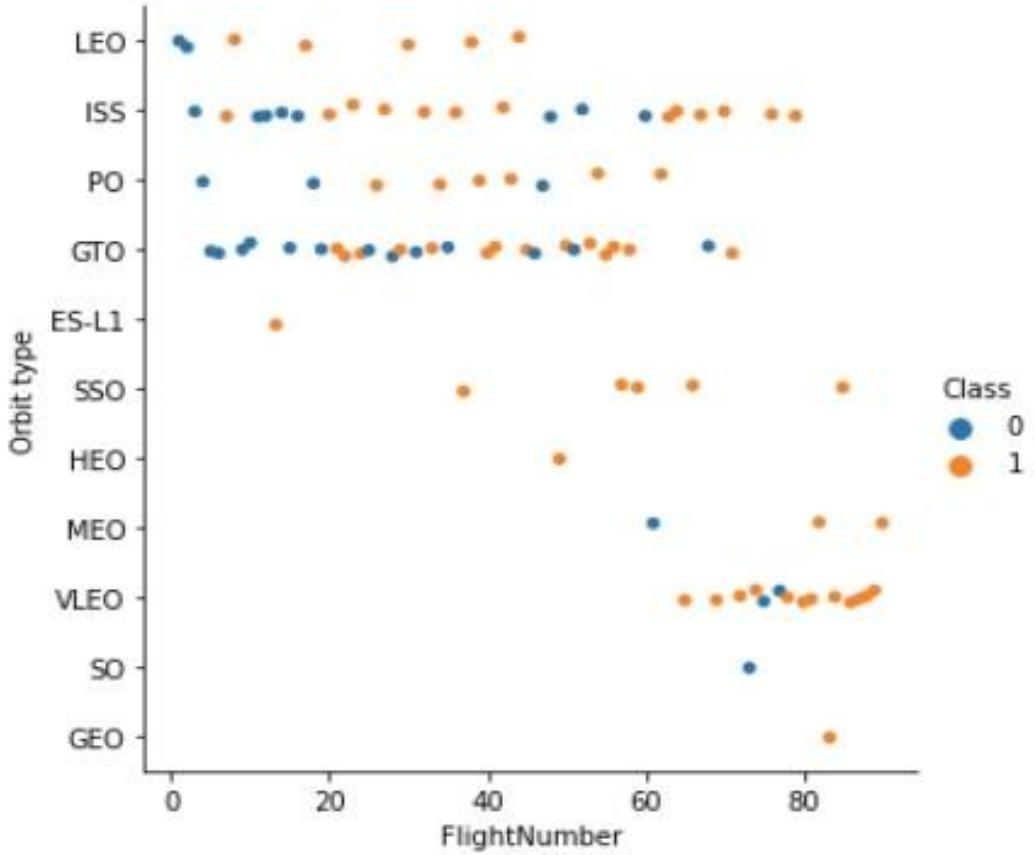
Flight Number vs. Orbit type

We can see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

We can also see that initially, only LEO, ISS, PO and GTO orbit missions were launched, while later SSO and VLEO orbits appear quite frequently.

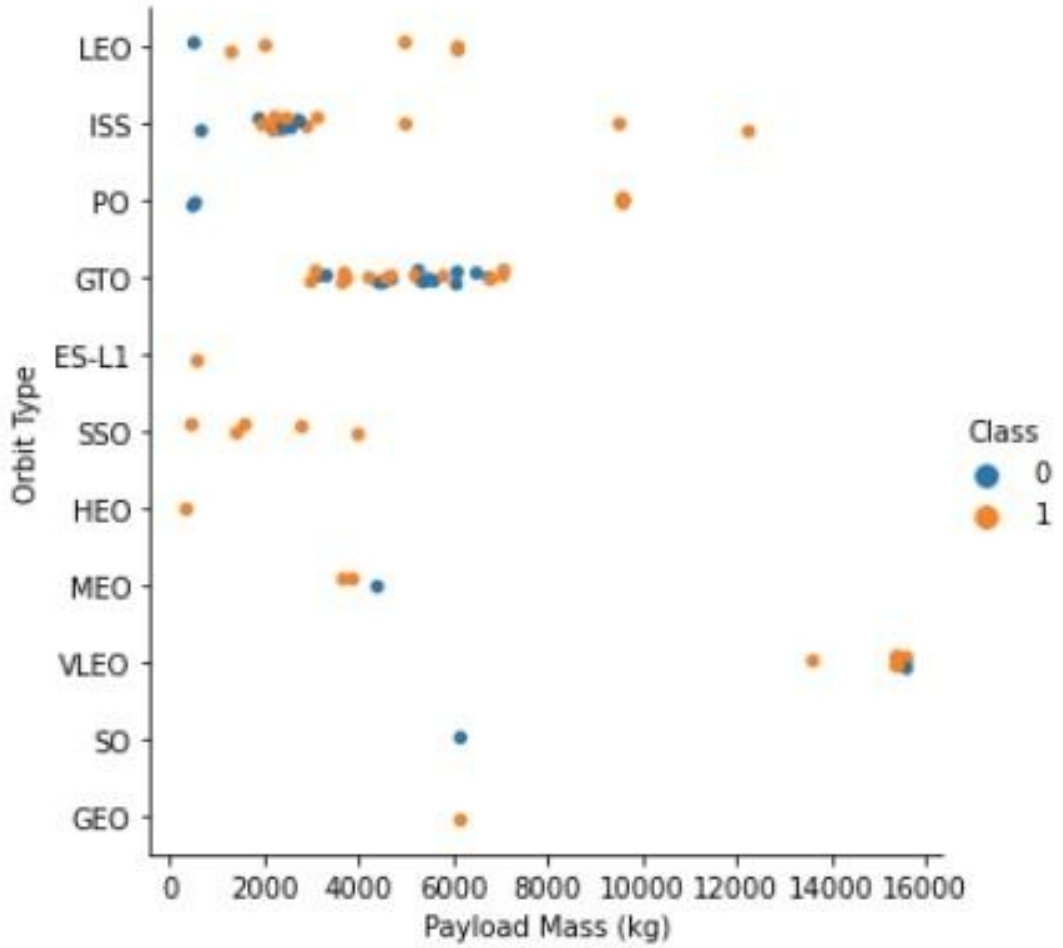
In contrast, only one mission was launched for each of ES-L1, HEO, SO and GEO orbits (explaining 100% and 0% scores from the previous slide). There were also only 3

```
sns.catplot(data=df,x='FlightNumber', y='Orbit', hue='Class')
plt.xlabel('FlightNumber')
plt.ylabel('Orbit type')
plt.show()
```



Payload vs. Orbit type

```
sns.catplot(data=df,x='PayloadMass', y='Orbit', hue='Class')
plt.xlabel('Payload Mass (kg)')
plt.ylabel('Orbit Type')
plt.show()
```



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

We can also see that VLEO orbit corresponds all of the heaviest launches.

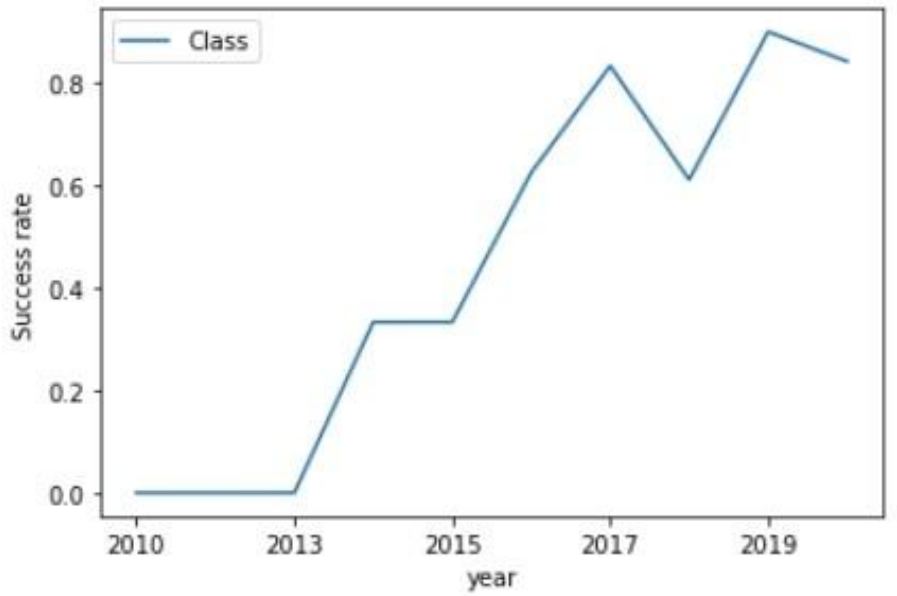
Launch success yearly trend

```
df_suc_per_year=df.groupby(Extract_year(1))['Class'].mean().reset_index()
df_suc_per_year.head()
```

]:

	index	Class
0	2010	0.000000
1	2012	0.000000
2	2013	0.000000
3	2014	0.333333
4	2015	0.333333

```
# Plot a line chart with x axis to be the extracted year and y axis to be th
df_suc_per_year.plot(x='index', y='Class')
plt.xlabel('year')
plt.ylabel('Success rate')
plt.show()
```



We can observe that the success rate since 2013 kept increasing till 2020 with a dip in 2018.

EDA with SQL

In subsequent slides we explore the dataset using SQL (through SQL magic in a jupyter notebook). In all of those, our dateset is stored in SPACEX table on ibm db2.

All launch site names

```
%sql select DISTINCT(LAUNCH_SITE) from SPACEX
```

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

We see that there are 5 distinct sites.
Two of them (#2 and #3) differ only by a space, so likely are just different representations of the same site.

Launch site names begin with `CCA`

```
%sql select * from SPACEX 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

We see that top 5 records all correspond to launches from CCAFS LC-40, but from the previous slide we know that there are launches from CCAFS SLC-40 as well.

Total payload mass launched by NASA (CRS)

```
%sql select SUM(PAYLOAD_MASS_KG_) from SPACEX where CUSTOMER='NASA (CRS)'
```

1
45596

We see that the total payload payload mass launched by NASA (CRS) is 45596 kg.

Average payload mass by F9 v1.1

```
%sql select AVG(PAYLOAD_MASS_KG_) from SPACEX where BOOSTER_VERSION='F9 v1.1'
```

1
2928

We see that the average payload mass carried by booster version F9 v1.1 is 2928 kg.
In the query above, condition 'where' makes sure that we only average those payload masses that have the correct booster version.

First successful ground landing date

```
%sql select MIN(DATE) from SPACEX where
LANDING__OUTCOME='Success (ground pad)'
```

1
2015-12-22

We see that the first successful ground landing happened for a rocket launched on 22nd of December of 2015.

Successful drone ship landing with payload between 4000 and 6000

```
%sql select DISTINCT(BOOSTER_VERSION) from SPACEX where
(LANDING__OUTCOME = 'Success (drone ship)') and
(PAYLOAD_MASS_KG_<6000) and (PAYLOAD_MASS_KG_>4000)
```

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

We see that there are 4 versions of boosters that satisfy all conditions.

Note that the first condition is somewhat confusingly formulated, so there might be several other possible queries, e.g. replacing "LANDING__OUTCOME = 'Success (drone ship)'" with
"LANDING__OUTCOME like 'Success%'" would return all successful landings.

Total number of successful and failure mission outcomes

```
%sql select COUNT(*),MISSION_OUTCOME from SPACEX
group by MISSION_OUTCOME
```

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

We see that vast majority (99 out of 101) missions were successful.

One mission was a general success, but the payload status is unknown due to limited information available, see https://en.wikipedia.org/wiki/Zuma_(satellite)

One mission, SpaceX CRS-7, have failed, pictured to the left.
See Wikipedia article for details.
https://en.wikipedia.org/wiki/SpaceX_CRS-7

Boosters carried maximum payload

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

```
%sql select BOOSTER_VERSION from SPACEX
where PAYLOAD_MASS_KG_=(select
MAX(PAYLOAD_MASS_KG_) from SPACEX)
```

We see that all of the boosters that carried maximal load are variations of F9 B5 booster. Indeed, this is the most up-to-date booster version Falcon 9 Block 5: https://en.wikipedia.org/wiki/Falcon_9_Block_5

2015 launch records

```
%sql select MONTHNAME(DATE) as month,
LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from
SPACEX where (YEAR(DATE)=2015) and (LANDING__OUTCOME
like '%Failure%')
```

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

We see that there are only 2 failure records in 2015: one in January, one in April. Both of the failures were drone ship failures.

Note that again, the formulation of the question is not very precise, if we want all records, and not just the ones with failed landings, we would remove the LANDING__OUTCOME like '%Failure%' condition.

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

```
%sql select count(*) as
number_of_successfull_landings,LANDING__OUTCOME from
SPACEX where (DATE>='2010-06-04' and
DATE<='2017-03-20' and LANDING__OUTCOME like
'Success%') group by LANDING__OUTCOME order by
number_of_successfull_landings desc;
```

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

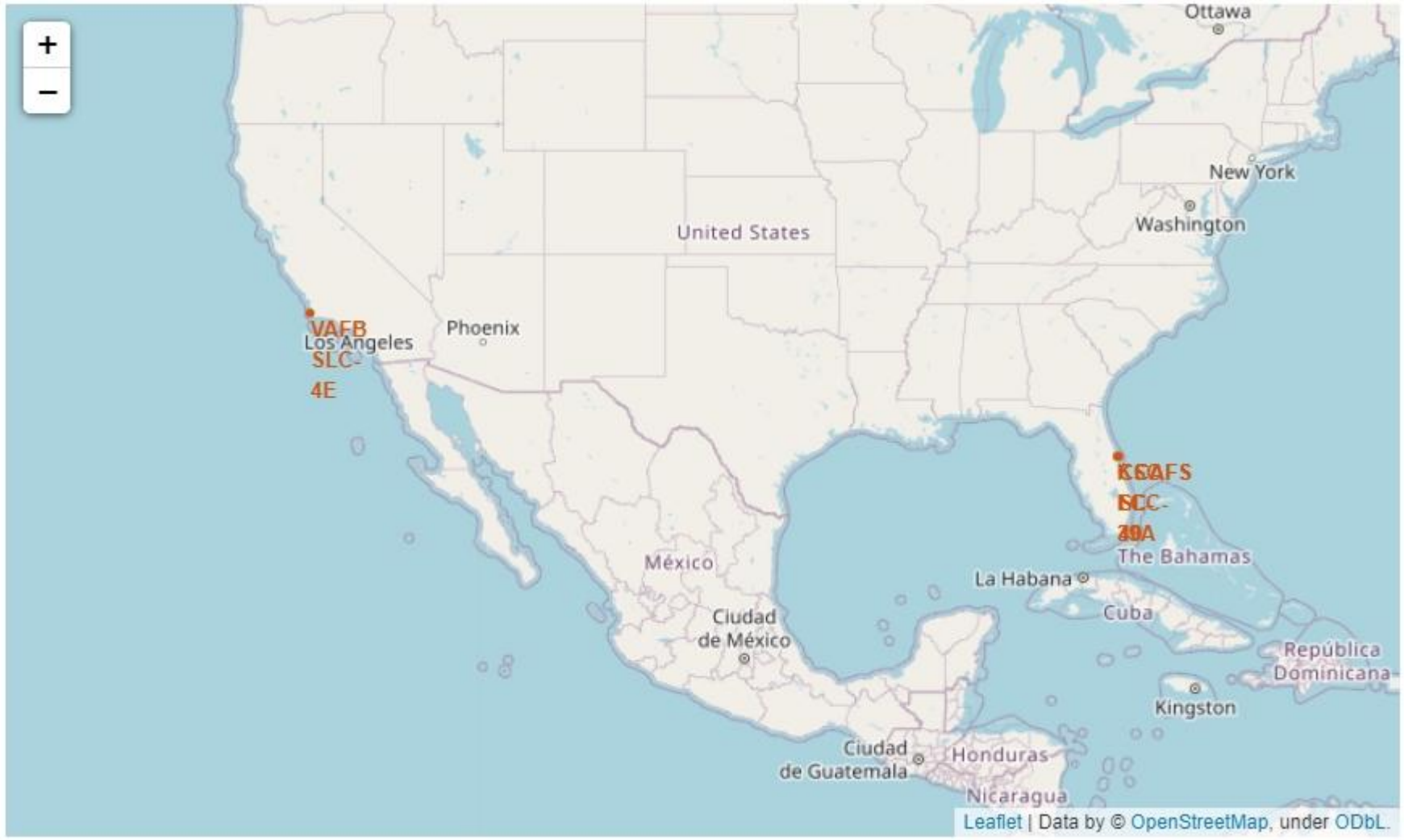
We see that there are 5 drone ship successes and 3 ground pad successes.
Again, the question is unclear as it doesn't specify how to group and rank the counts, but it is easily adjustable.

Interactive map with Folium

All of the maps are interactive when viewed in a jupyter notebook. It is available here:

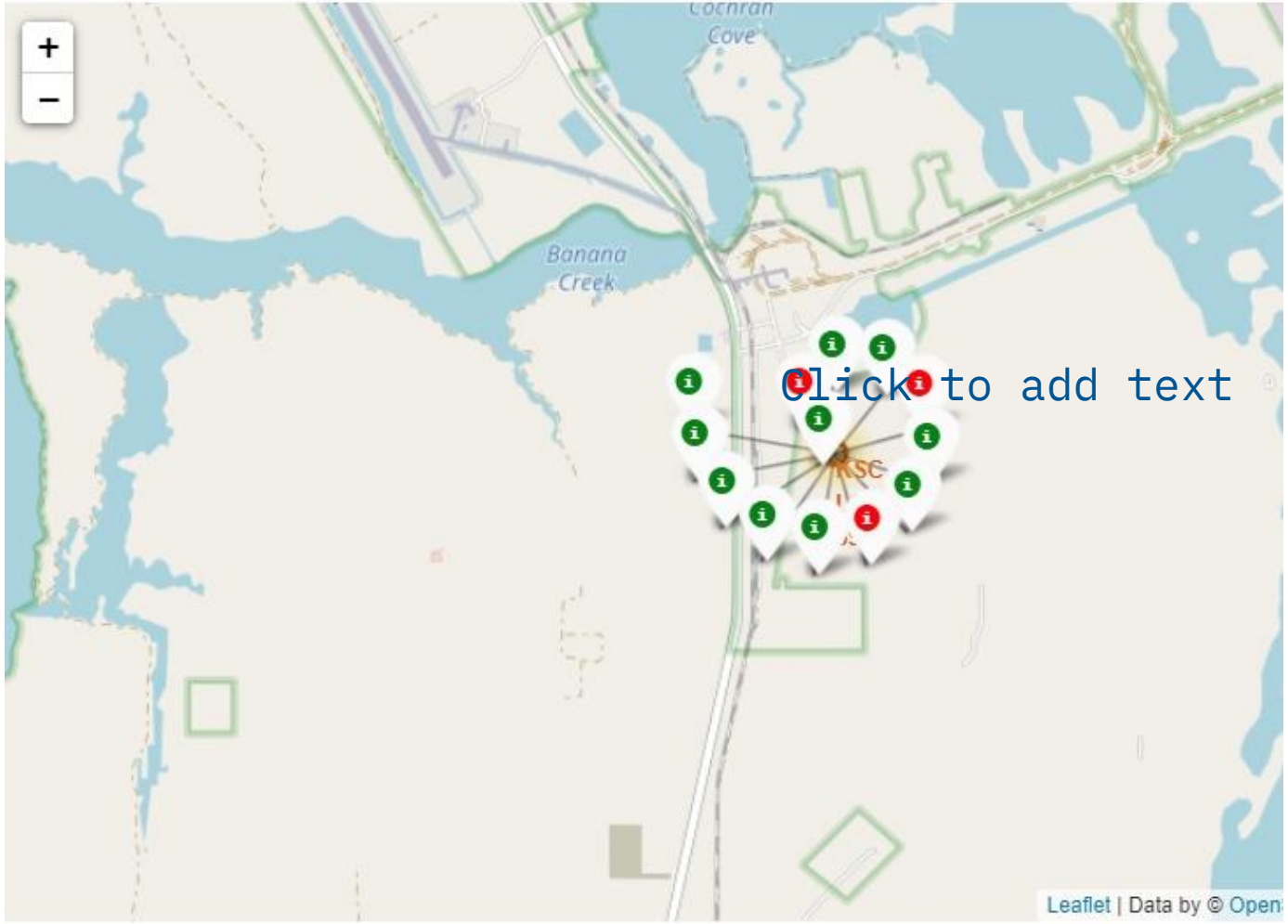
https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/cf9b59b8-29ec-4767-8106-10b8ec293740/view?access_token=55e09764677688a61ae925448ee5deb353a3e096ef09d321ebbc38f13f31c08f

Folium map with marked launch sites



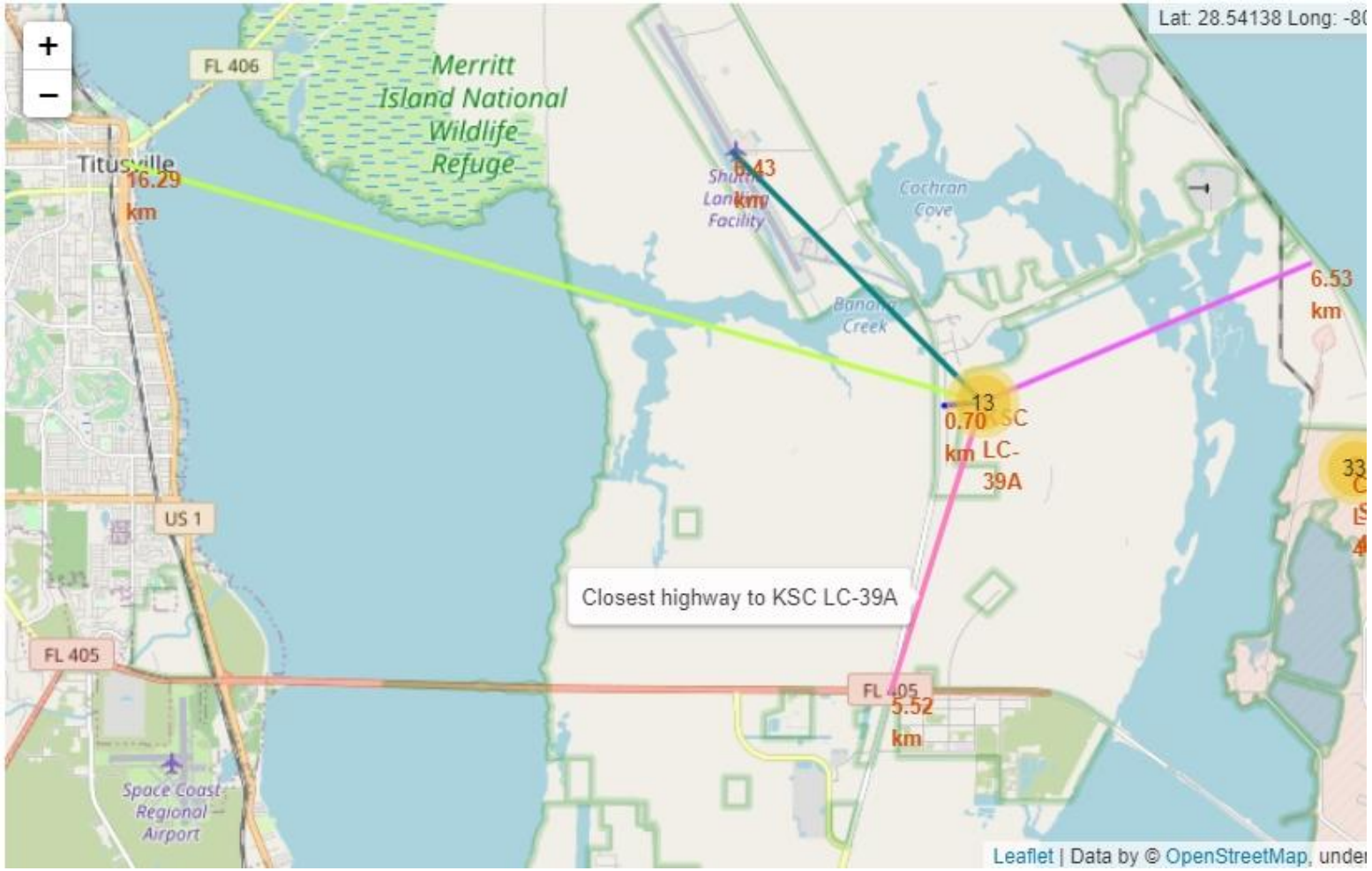
We see that 3 out of 4 sites are located in Florida right next to each other, and one site is on the West coast. We also see that all sites are located by the ocean.

Folium map displaying all launches from a site



This screenshot illustrates the number and successes of launches from a specific site (here displayed KSC LC-39A).
Green tags correspond to successful outcomes, red to unsuccessful.
We noticed that most of launches from this site were successful.

Folium map displaying distances to proximities



This map illustrates distances to various vital objects from a specific site (KSC LC-39A).
Objects included are:

a) Highway
b) City
c) Airport
d) Coast line
e) Rail station

Each object shows a tag with the distance from that object to the site. In Jupyter notebook we can also hover over each line to display a tooltip, as shown here for the highway line.

We can see that site KSC LC-39A is relatively close to all important objects.

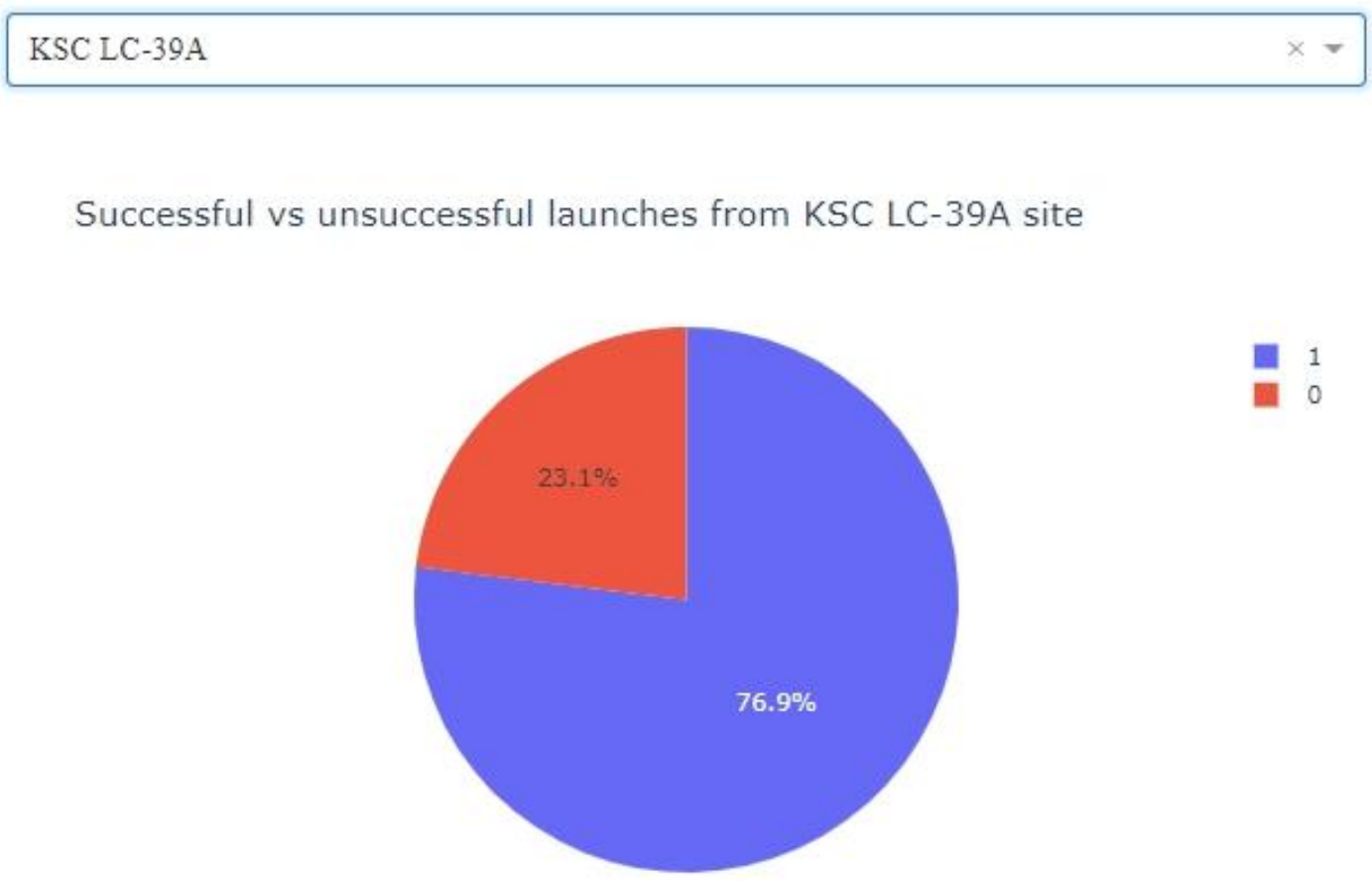
Build a Dashboard with Plotly Dash

Here we describe a dash-built dashboard that illustrates some other insights into the data. The code is available at https://github.com/DenisOvchinnikov93/IBM-data-science-capstone/blob/master/main.py

Total successful launches per site

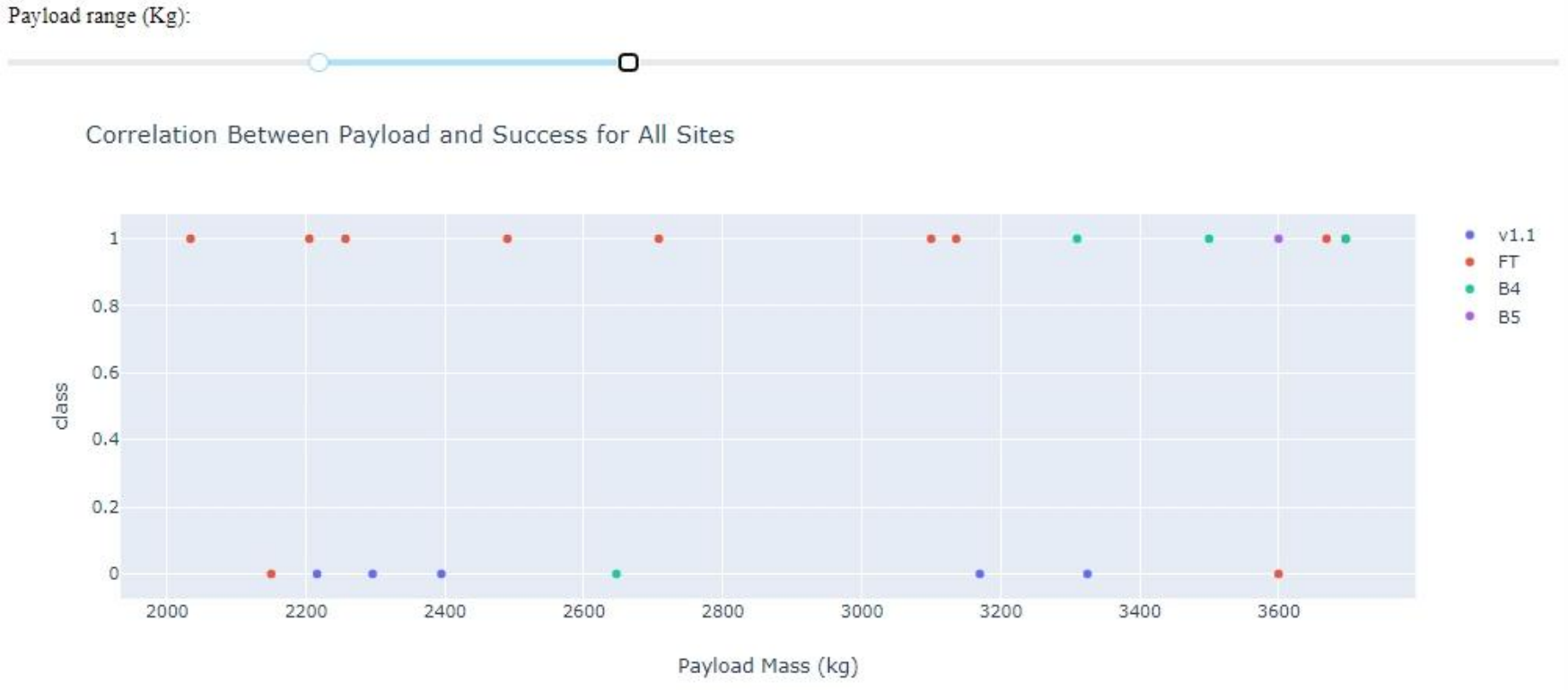
This piechart allows us to examine the number of successful launches from each site. For example it is evident that KSC LC-39A has the most launches, while CCAFS SLC-40 has the least number of launches.

Success rate for KSC LC-39A



Upon inspecting each site, we see that KSC LC-39A has the highest success rate of 76.9%

Scatter plot of payload vs success



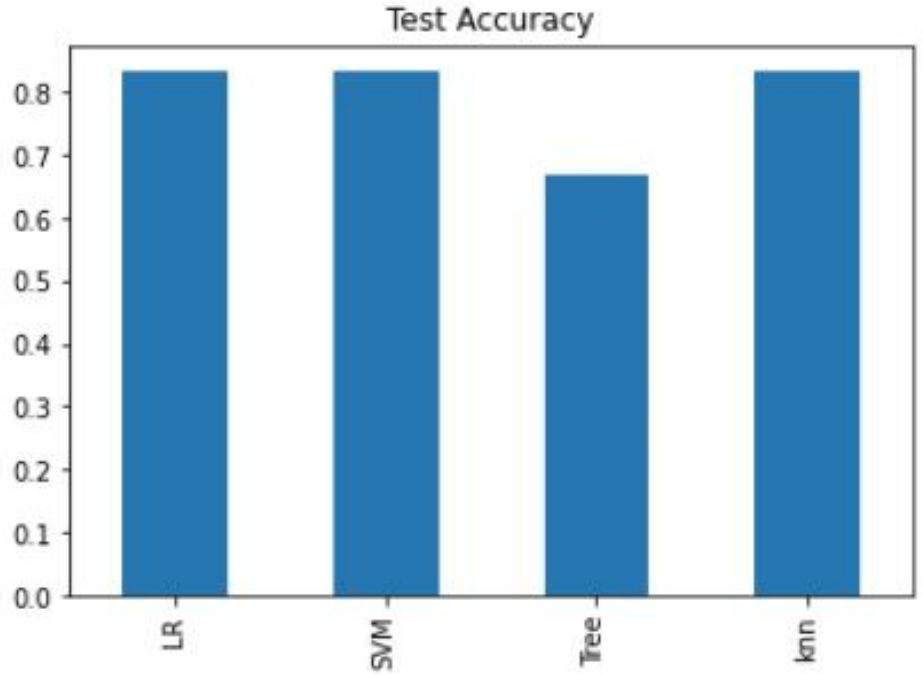
Each point is colored according to the booster version.

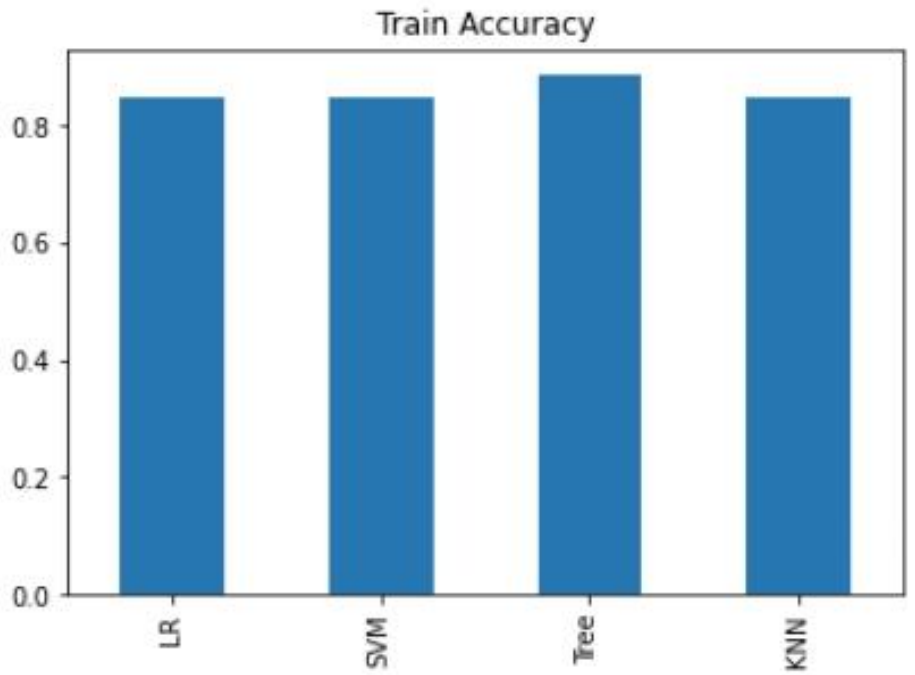
We see that in the payload mass range of 2000-3800 kg, most of FT boosters (corresponding to red-colored points) were successful. We can also see that most of v1.1 bosster launches (corresponding to blue points) were unsuccessful.

Predictive analysis (Classification)

Here we summarize our predictive analysis. Details can be found in the notebook: https://github.com/DenisOvchinnikov93/IBM-data-science-capstone/blob/2f96b99dc9a7a1d1f0b00002e4c846944c656988/capstone%20week%204-%20Machine%20Learning%20Prediction.ipynb

Classification Accuracy





We see that on a test set, Logistic Regression, SVM and KNN perform the same.

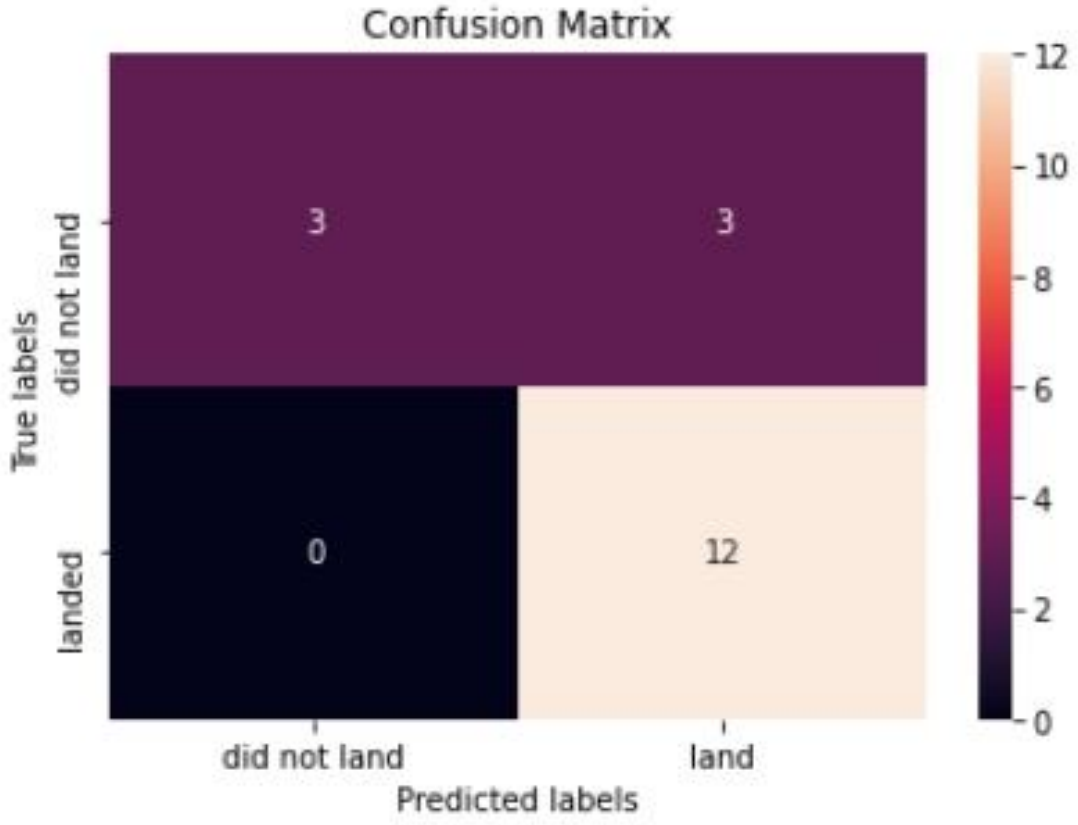
They also perform very similarly on the train set, so we can choose either of them. I will choose KNN due to it's interpretability.

Note that in the jupyter notebook provided on Coursera, there is a typo that uses svm_cv model to evaluate tree model performance, this way all models appear to have the same performance.

Confusion Matrix

The confusion matrix for the chosen model (KNN) is shown here. We see that the algorithm performs generally well, witht the only problem being the false positives.

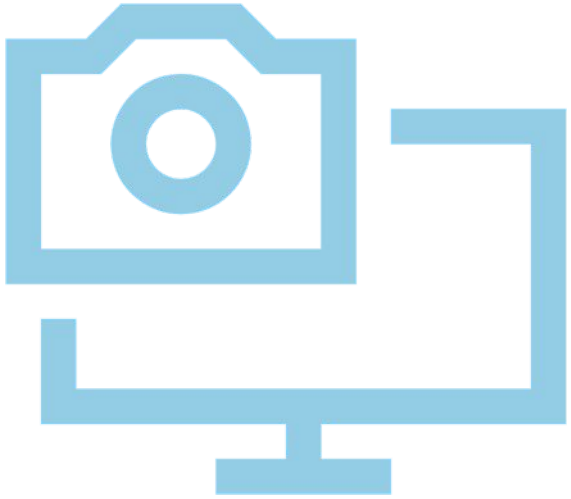
```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



CONCLUSION

- We established a pipeline to scrap, clean and process the data on Falcon-9 launches.
- We separated features that hav e an impact on whether a launch was successful or not.
- We built and tested several predictive models and chosen one of them to serve as our main predictor.

APPENDIX



https://github.com/DenisOvchinnikov93/IBM-data-science-capstone

https://github.com/DenisOvchinnikov93/IBM-data-science-capstone/blob/2f96b99dc9a7a1d1f0b00002e4c846944c656988/spacex_launch_dash.csv