IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anton Rechenauer

May 28th 2022

Outline

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

Executive Summary

- **Summary of methodologies**
  - **Data Collection classical style (CSV files)**
  - **Data Collection using web scraping (from Wikipedia)**
  - **Data Wrangling and Cleaning**
  - **Exploring the data - SQL, Visualizations, GIS(Folium)**
  - **Building an interactive dashboard**
  - **Prediction (here: Classification) using Machine Learning**
- **Summary of all results**
  - **Data Analysis results: success is more likely for light payload**
  - **Discussion of the quality of the predictions from ML – we can achieve an accuracy of more than 80%**

Introduction

Project background and context:

SpaceX made payload transport to an orbit cheap – they can reuse the first stage (aka “Stage One”) of their rocket Falcon9. This cuts down the cost from \$165m to \$62m per launch!

But there are still a lot of unsuccessful attempts to recover the first stage!

This raises an obvious question:

Can we find out the conditions or parameters for a successful recovery of Stage One? Is there a – explainable - reason why it sometimes fails?

A first baby step is to check whether we can predict success/failure!

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
  - Collect rocket launch data via REST-API from SpaceX, get data from web scraping on Wikipedia
- **Perform data wrangling**
  - Modify and add Columns, remove outliers, discuss missing data, One-Hot Encoding etc.
- **Perform exploratory data analysis (EDA) using visualization and SQL**
- **Perform interactive visual analytics using Folium and an interactive dashboard**
- **Perform predictive analysis using classification models**
  - We do the usual Test/Train split and then train several classification models to predict success or failure of recovery.
  - For each model the best hyperparameters are found using cross-validation
  - We finally compare the accuracy of these models using test data

Data Collection

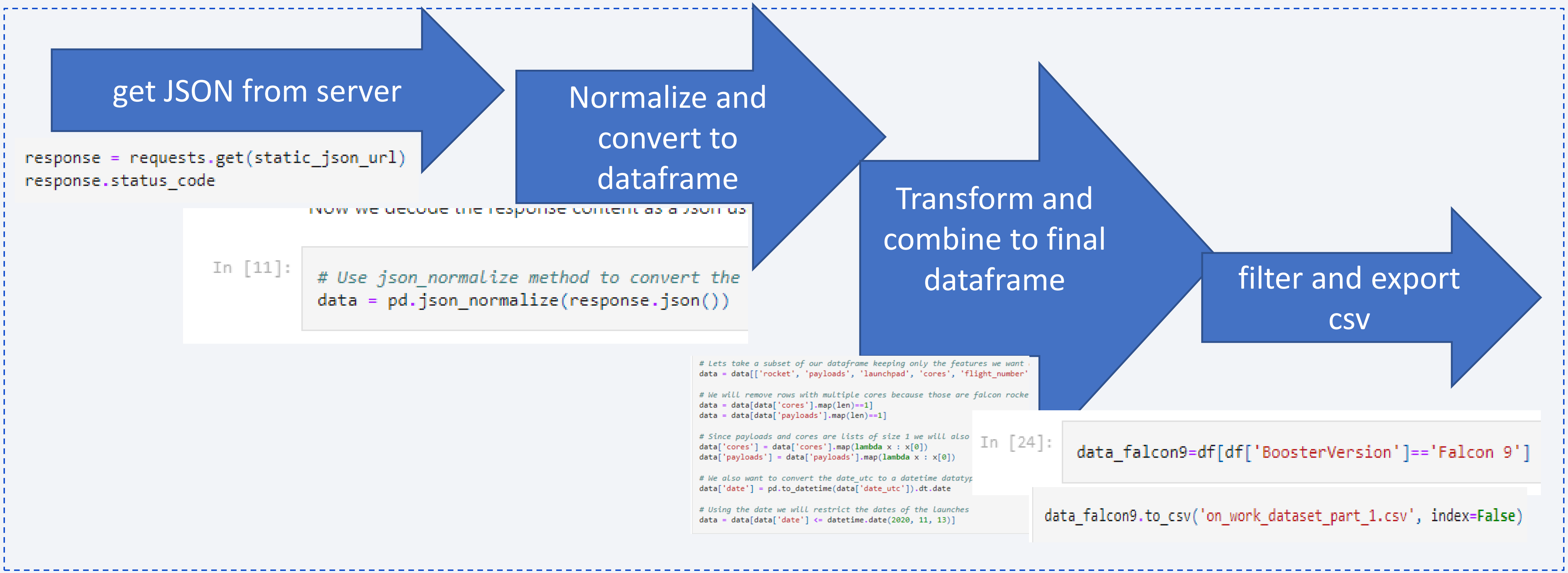
- **Data Collection “SpaceX”**
  SpaceX provides a REST-API where several JSON-files (data about the rockets, payloads, launchpad, cores) files could be downloaded. These where combined, filtered for the falcon9 data and then exported in csv format.

- **Data Collection “Wikipedia”**
  Python libraries were used to retrieve the HTML of a relevant webpage (as text) and the launch related data was extracted using the famous library htmlsoup.

Data Collection – SpaceX API

GitHub Link

https://github.com/Bleiglanz/Capstone_SpaceX/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection – Web Scraping from Wikipedia

GitHub
https://github.com/Bleiglanz/Capstone_SpaceX/blob/main/jupyter-labs-webscraping.ipynb

get HTML text

```
res = requests.get(static_url)
res
```

soupify

```
soup = BeautifulSoup(res.text,"html5lib")
```

Get tables and columns

```
# Assign the result to a list called  h
html_tables = soup.find_all('table')
```

Starting from the third table is our target table

```
from IPython.display import HTML
first_launch_table = html_tables[2]
```

Build a dictionary from columns

```
launch_site = row[2].a.string
launch_dict['Launch site'].append(''+launch_site)

payload = row[3].a.string
launch_dict['Payload'].append(''+payload)

payload_mass = get_mass(row[4])
launch_dict['Payload mass'].append(payload_mass)
```

Transform

Dictionary to dataframe

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

Export to csv

```
df.to_csv('owncsv/own_work_spacex_web_scraped.csv', index=False)
```

Data Wrangling

- **Exploratory Data Analysis (EDA) is the process of cleaning, unifying and understanding complex data.**
- **The goal is insight into the data, it’s structure and discovery possible problems.**

In this project:

- Find number of launches at each site
- Find the orbit for each launch
- Work out the success rates (Labels)

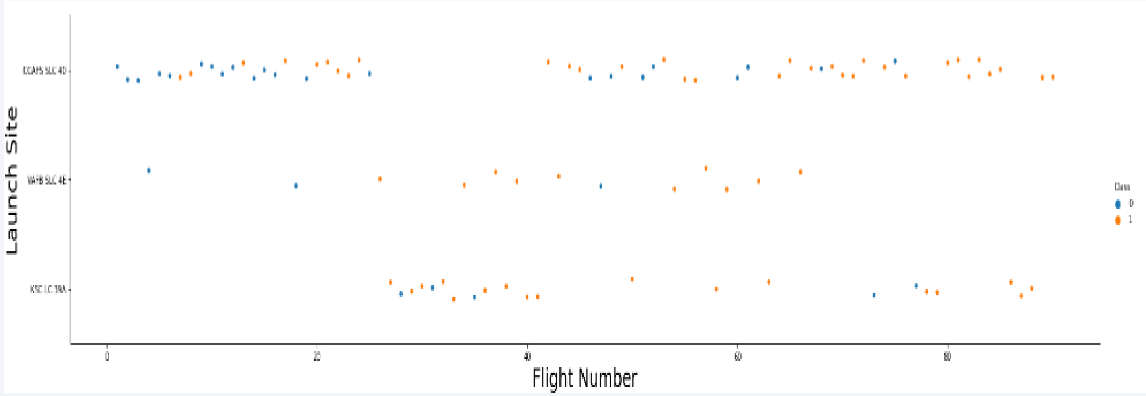
GitHub
https://github.com/Bleiglanz/Capstone_SpaceX/blob/main/labs-jupyter-spacex-Data-wrangling.ipynb

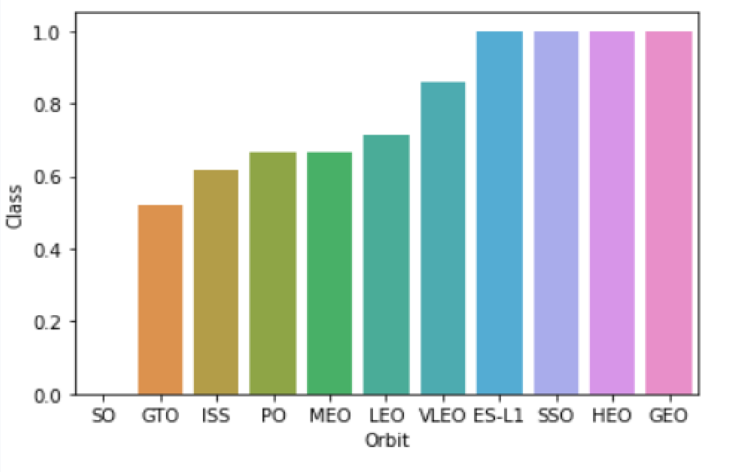
EDA with Data Visualization

GitHub:

https://github.com/Bleiglanz/Capstone_SpaceX/blob/main/jupyter-labs-eda-dataviz.ipynb

Examples: flight vs. launch site, success vs. orbit





The following scatter plots were created:

Payload Mass / Flight Number **Launch Site / Flight Number** **Launch Site / Payload Mass** **Orbit / Flight Number,** **Playload / Orbit**

In addition, we created a bar graph **(Success Rate / Orbit)** and a Line Graph **(Success Rate / Year)**

EDA with SQL: We performed several queries

GITHUB
https://github.com/Bleiglanz/Capstone_SpaceX/blob/main/jupyter-labs-eda-sql-coursera.ipynb

- Names of unique launch sites
- Five records where launch sites begin with CCA
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster F9 v1.1.
- Date of first successful landing outcome in ground pad
- Name of boosters successfully recovered in drone ship and have payload mass >= 4000 and <= 6000.
- Count of successful and failure mission outcomes.
- Names of the Boosters of maximum payload mass.
- List the records which will display the month names, failures in drone ship, booster versions in 2015.
- Rank successful landing between the date 04-06-2010 and 20-03-2017 in descending order.

Hint to grader: My Watson account was blocked, I used sqlite!

Interactive Map with Folium

GitHub
https://github.com/Bleiglanz/Capstone_SpaceX/blob/main/lab_jupyter_launch_site_location_folium.ipynb

- We used Folium to add Markers, Circles and Lines to an interactive map showing the launch sites of the Falcon9 rocket
- Markers indicate points (here: launch sites) and the frequency of launches at each site (green=successful recovery of Stage One for this launch, red=failure of recovery)
- Circles were used to highlight areas
- Lines were used to indicate distances (closest proximity to coastline, city, railway etc)

Dashboard with Plotly Dash

GitHub
https://github.com/Bleiglanz/Capstone_SpaceX/blob/main/dashboard/spacex_dash_app.py

The dashboard contains to graphs:

A **pie chart** showing the total number of launches by a site (or all of them combined). For a selected launch site, the pie chart showed a breakdown of launches (successful/failures)

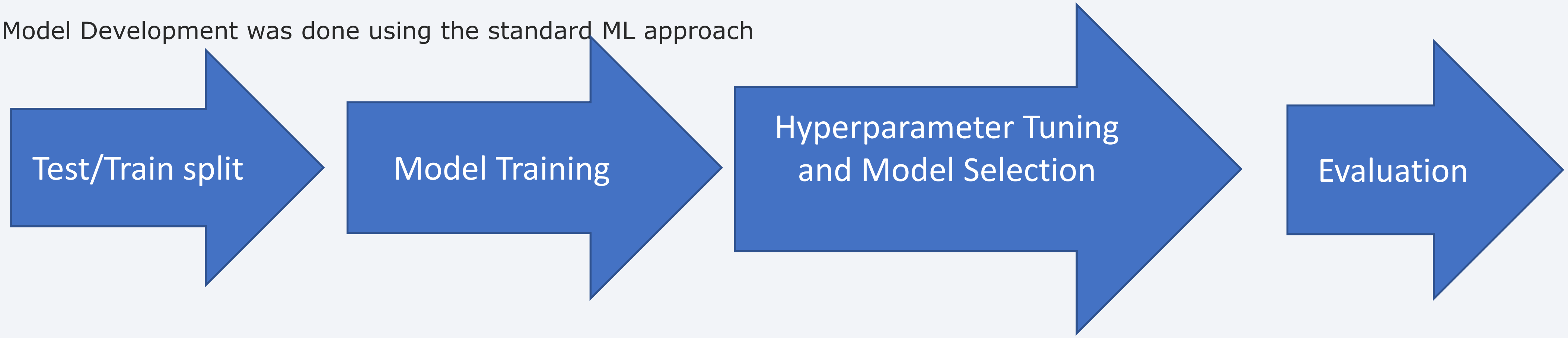
A **scatter plot** of the outcome (successful recovery of Stage One or not) versus the payload mass. The bound of the mass can be interactively changed by the user.

Predictive Analysis (Classification)

GitHub

https://github.com/Bleiglanz/Capstone_SpaceX/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5_Prediction.ipynb

Model Development was done using the standard ML approach



In the Lab we used scikitlearn GridSearchCV find the best hyperparameters for a variety of models (Logistic Regression, SVM, Decision Tree, KNN). For every model the confusion matrix was created, and we compared the models using a simple score (accuracy).

Since the dataset was very small and not to skewed this seems ok. In a future analysis the F1-score should be used!!

Results

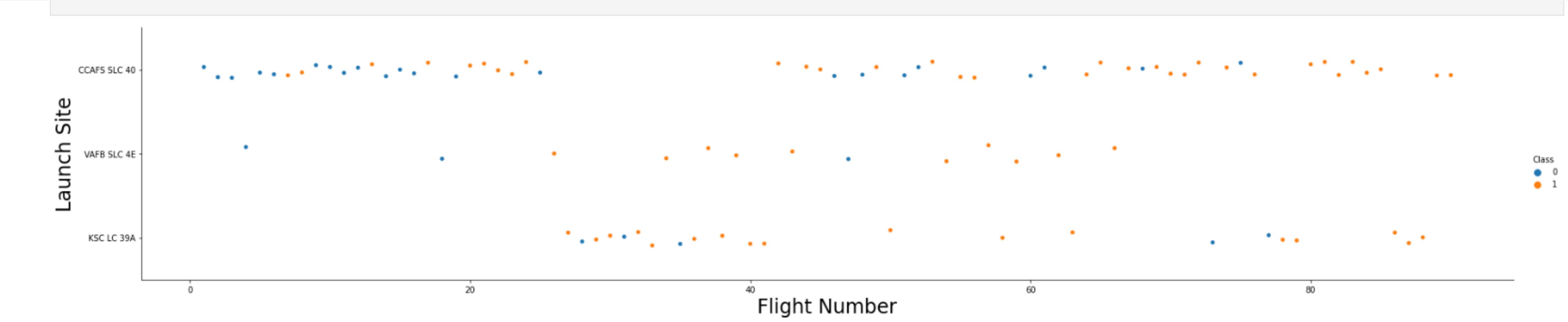
- From EDA we see that more recent starts have a better chance of successful booster recovery (obviously SpaceX learned how to do it better in the course of time)
- There is a baseline of about 80% success (SpaceX has already achieved this)
- Unsuccessful launches are in the past!
- One site stands out: KSC LC-39A has a success rate of 75%
- We suggest to look at drone ships: using these increases the chances of a successful recovery of the Stage One!
- It is possible to predict the results with a accuracy of 83%. But this means almost nothing – the dataset is very, very small and a bit skewed.

It is a small dataset, results a confounded by the technological progress SpaceX has made in the past few years. But let's look at the details:

Section 2

Insights drawn from EDA

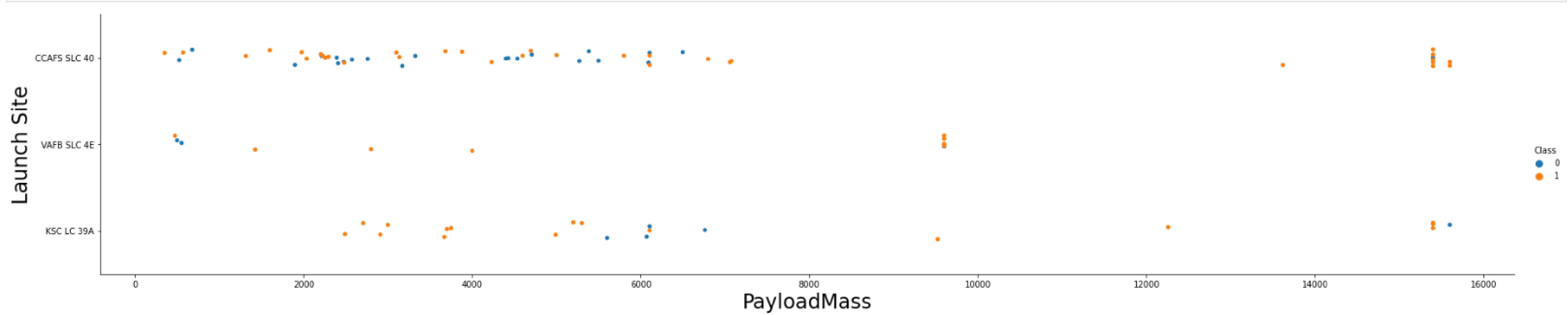
Flight Number vs. Launch Site



We see more and more orange=successful recoveries in the course of time (higher flight number).

This means that *at every site* the situation gets better, chances of success are increasing!

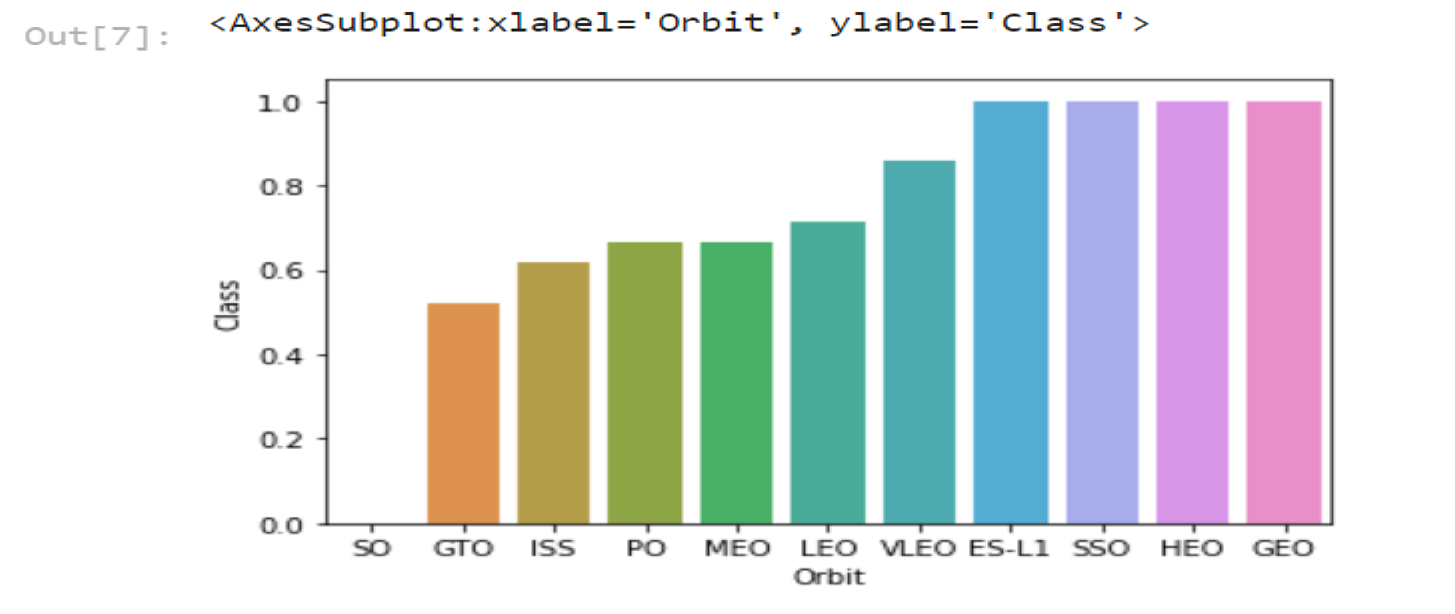
Payload vs. Launch Site



The highest chance of a successful recovery is a mid-range or high payload.

But this effect isn't very convincing.

Success Rate vs. Orbit Type



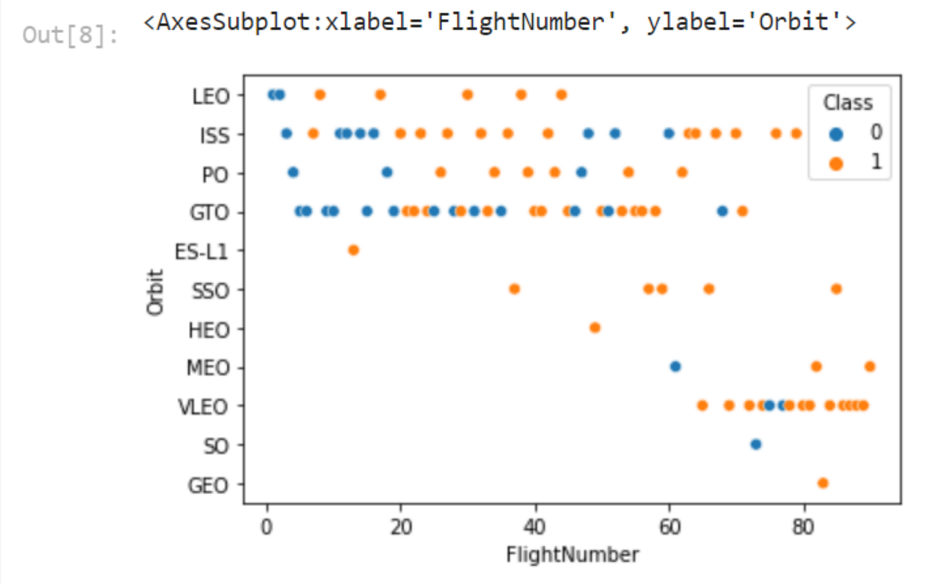
- It seems obvious that some orbits are perfect for success (100% recovery rate).

  But GEO has only one launch – be careful here, use absolute numbers to be sure!

- This chart **is not** an explanation of success/failure in the whole dataset. The successful orbits are the ones tried *later* in time (after SpaceX learned how to do it)

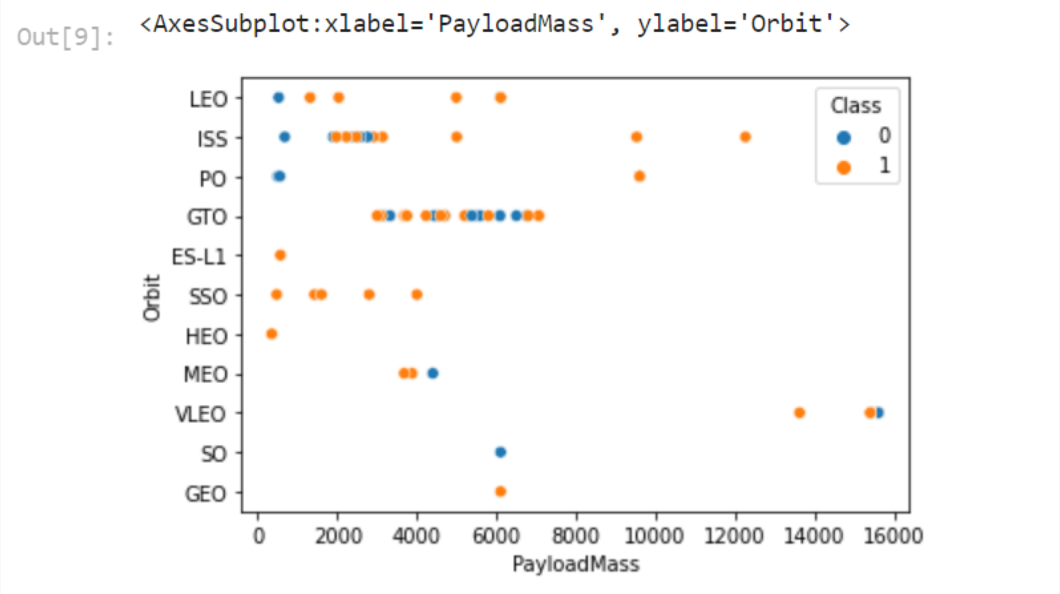
- Nethertheless the orbit (height, involved velocity etc.) could affect the success rate very much.

Flight Number vs. Orbit Type



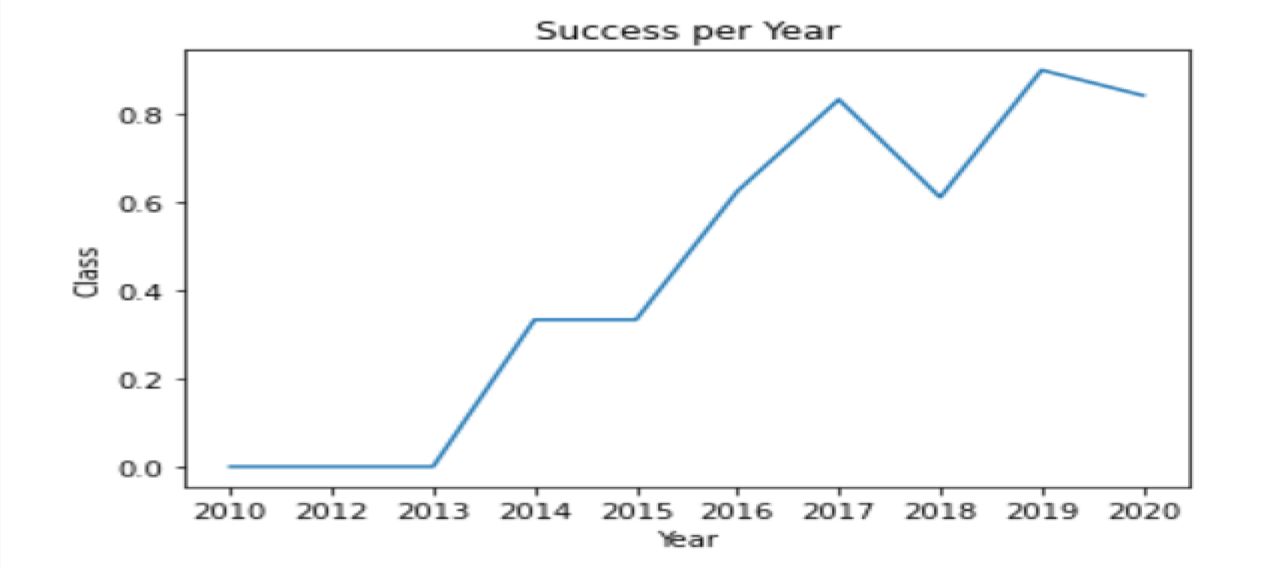
- This chart should be compared to the previous one
- Not every orbit was in the program in the past!
- The GEO orbit is possbly an outlier (maybe it should be removed from the dataset)

Payload vs. Orbit Type



- There is no relevant relation between orbit type and payload
- GTO has a narrow, continuous range of payloads, ISS a wide one
- We see GEO and SO as possible outliers again

Launch Success Yearly Trend



- The success rate is increasing over the years
- SpaceX has achieved almost 80% success rate now
  (so the accuracy of our prediction from the ML model is questionable)

All Launch Site Names

- Find the names of the unique launch sites:

```
%sql select distinct "Launch_Site" from spacextbl
```

```
 * sqlite:///spacex.db
Done.
```

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

- "**distinct**" removes duplicates from the resultset.

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

```
%sql select * from spacextbl where "Launch_Site" like 'CCA%' limit 5
```

```
 * sqlite:///spacex.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677

- The clause like '*pattern*%' finds all string beginning with *pattern*
- Limit 5 displays only the first five rows
- result is not reproducible, since there might be more than five rows (and the resultset is unordered!)

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql select sum("PAYLOAD_MASS__KG_") from spacextbl where "Customer"='NASA (CRS)'
```

```
 * sqlite:///spacex.db
Done.
```

sum("PAYLOAD_MASS__KG_")
45596

- Ambiguous formulation: “from NASA” – there is only one customer like NASA

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

```
%sql select avg("PAYLOAD_MASS__KG_") from spacextbl where "Booster_Version"='F9 v1.1'
```

```
 * sqlite:///spacex.db
Done.
```

avg("PAYLOAD_MASS__KG_")
2928.4

- Simple query using a aggregation function in sql, in this case AVG
- The where clause filters exactly in this case

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome

```
In [61]: %sql select "Date" from spacextbl where "Landing _Outcome" like '%Success%pad%'  order by substr("Date",7,4),substr("Date",4,2),substr("Date",1,2)

 * sqlite:///spacex.db
Done.
```

Out[61]:

Date
22-12-2015

- This was more complicated in sqlite, because of weak date and time functionality
- In DB2 one would use something simple:

SELECT MIN(DATE) **FROM SPACETBL WHERE** LANDING_OUTCOME **LIKE** ‘%Success%pad%’

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

```
%%sql
select distinct "Booster_Version"
from spacextbl
where cast("PAYLOAD_MASS__KG_" as int) between 4000 and 6000
and "Landing _Outcome"='Success (drone ship)'
```

```
 * sqlite:///spacex.db
Done.
```

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

- Simple sql query using BETWEEN, also the question requires a DISTINCT

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%%sql
select
case when "Mission_Outcome" like '%Success%' then 'SUCC' else 'FAIL' end, count(*)
from spacextbl group by 1
```

```
 * sqlite:///spacex.db
Done.
```

case when "Mission_Outcome" like '%Success%' then 'SUCC' else 'FAIL' end	count(*)
FAIL	1
SUCC	100

- **Success** should be a Boolean field in the database, it is just bad practice to use like patterns

Boosters Carried Maximum Payload

- **List the names of the booster which have carried the maximum payload mass**
- One has to use a subquery
- First find the maximum payload
- Return this value from the subquery and use it in a where clause
- Since the same Booster could be used in several launches => DISTINCT is required

```
%%sql
select distinct "Booster_Version"
from spacextbl where cast("PAYLOAD_MASS__KG_" as int)
=(select max (cast("PAYLOAD_MASS__KG_" as int)) from spacextbl)
```

```
 * sqlite:///spacex.db
Done.
```

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

2015 Launch Records

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

```
%%sql
select "Landing _Outcome","Booster_Version","Launch_Site"
from spacextbl
where "Landing _Outcome" like '%Failur%drone%' and "Date" like '%2015%'
```

```
 * sqlite:///spacex.db
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

- In DB2 one should use YEAR(Date)=2015 of course

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
- This is hard to do in sqlite
- In DB2 it is easy, just the same query but with
  a better condition in the where clause:

“DATE” >= ’04-06-2010’ AND DATE<=‘20-03-2017’

```
%%sql
select "Landing _Outcome", count(*)
from spacextbl
where cast(substr("Date",7,4) as int)
between 2010 and 2916 group by 1 order by 2 desc
```

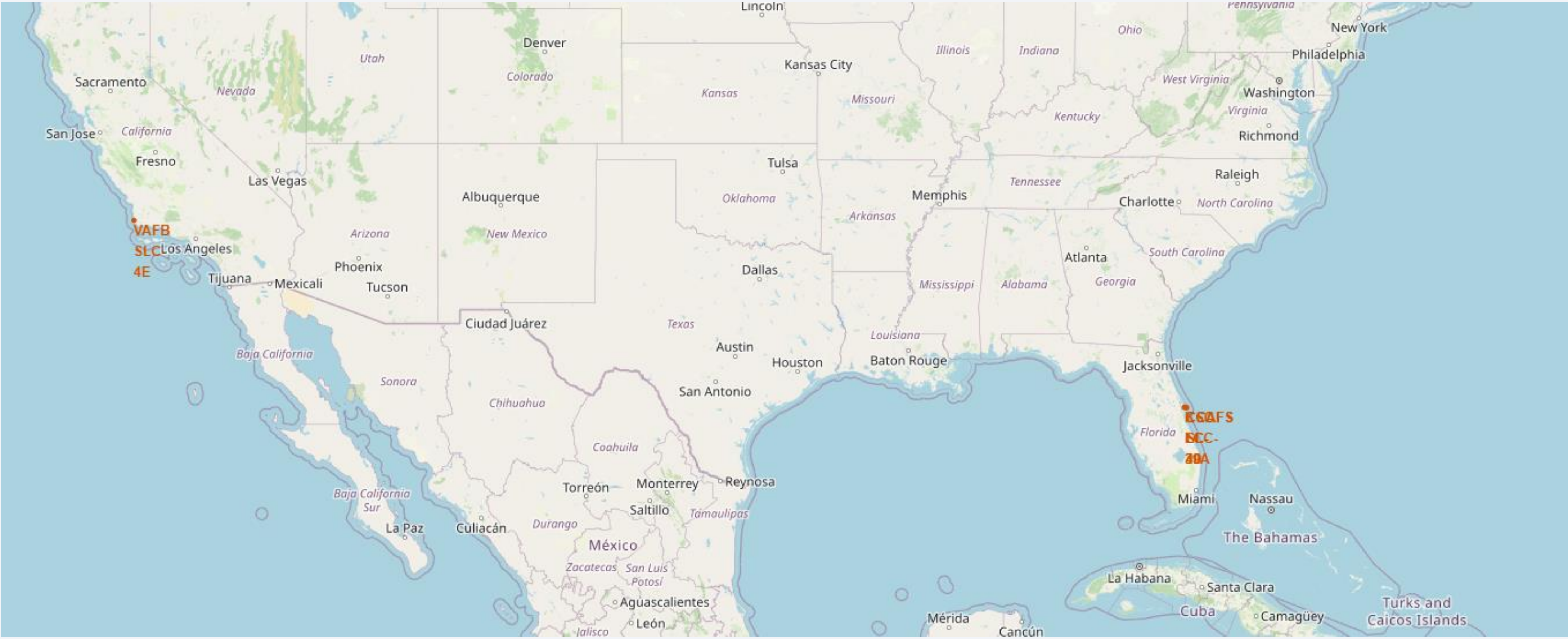
```
 * sqlite:///spacex.db
Done.
```

Landing _Outcome	count(*)
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

Section 3

Launch Sites Proximities Analysis

All Launch Sites of SpaceX



- Just showing the launch sites on a map of the USA

Icons and Markers – Launches

Each marker shows a launch.

The color coding for the markers is

Green = Success, Booster recovered

Red = Failure, Booster not recovered

Distances

Lines can be used to show distances

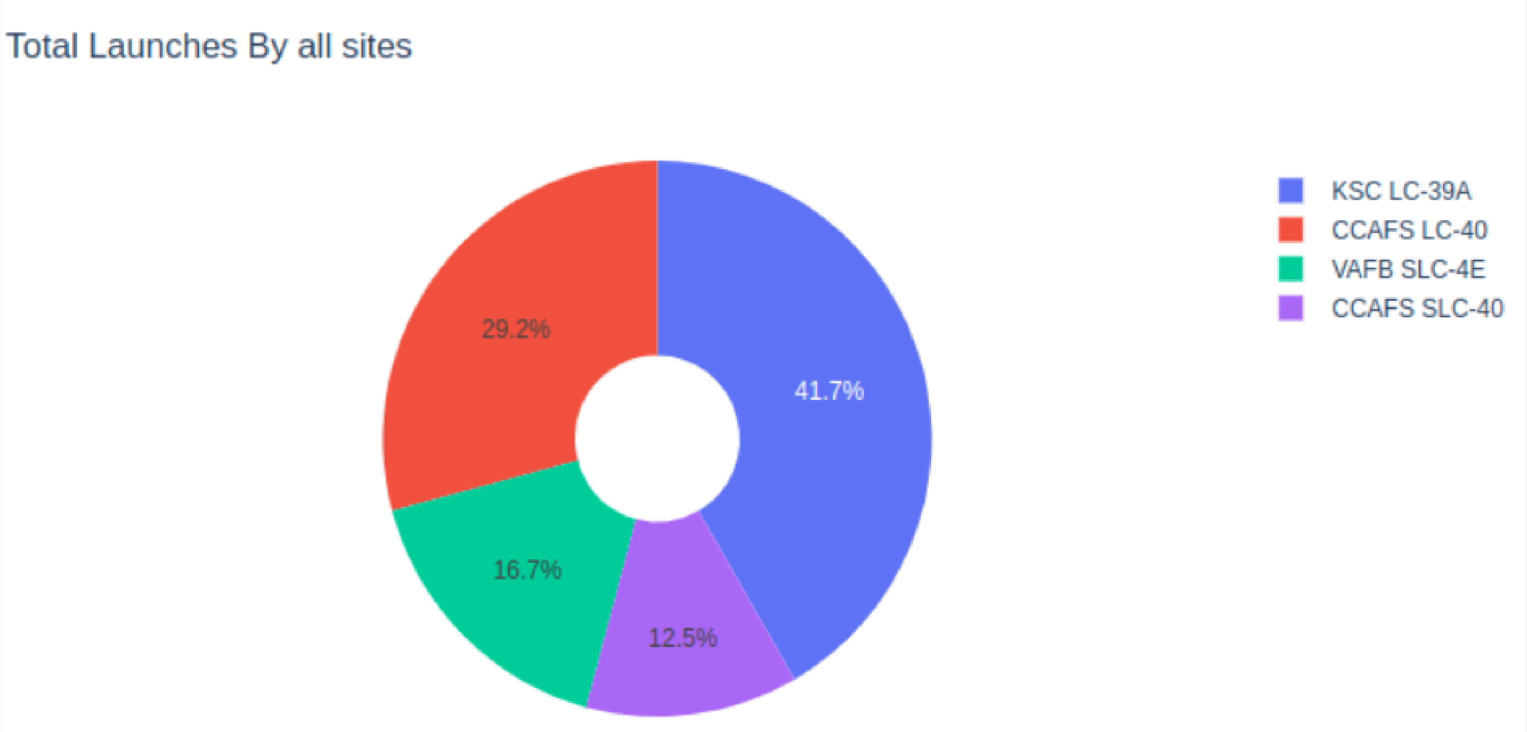
For example:

How close is the launch site to the coastline?

Section 4

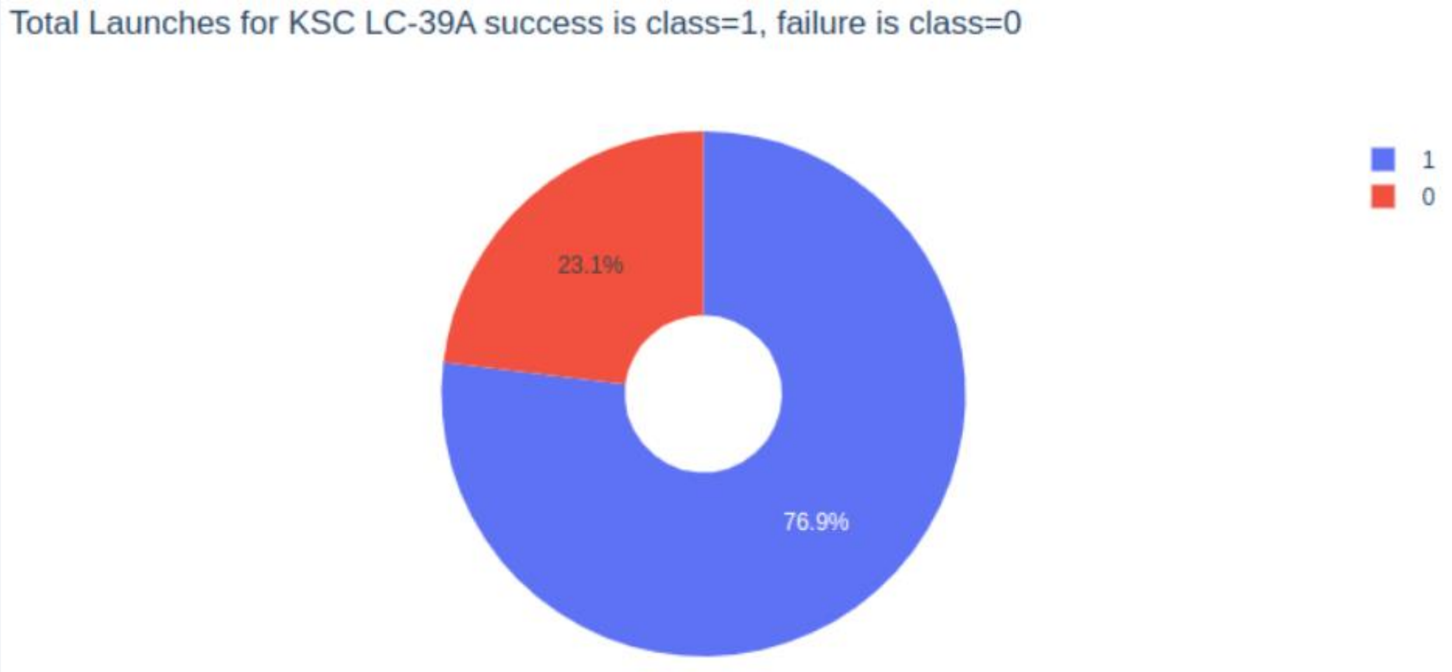
Build a Dashboard with Plotly Dash

Pie Chart – Success percentage per Site



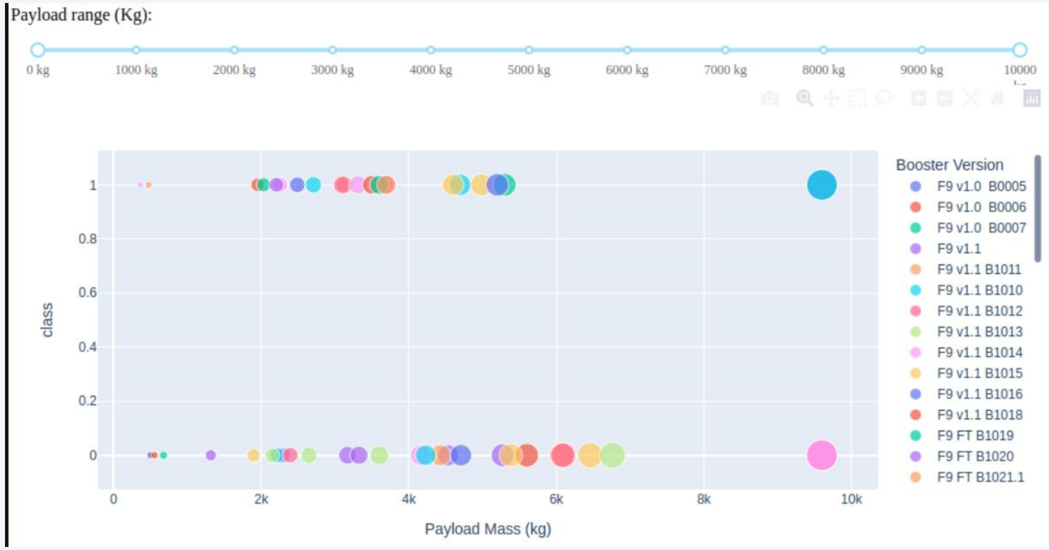
- KSC has the highest success rate for all sites

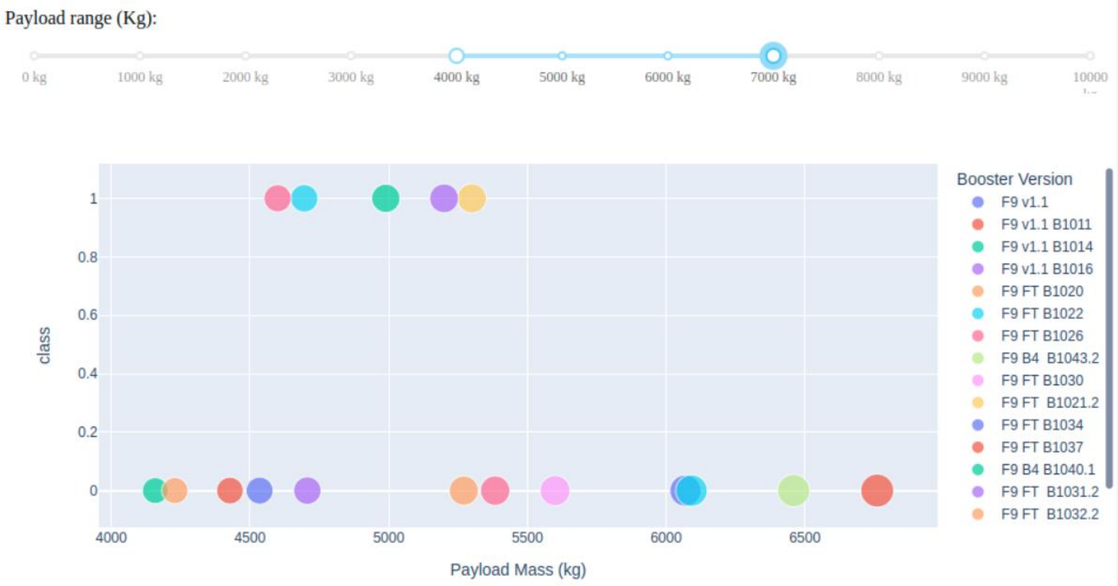
Pie Chart – Launch Site with highest launch success ratio



- Selecting a site shows the distribution of failures/success for each site
- KSC has the highest percentage of successful recoveries of Stage One Booster

Scatterplot of payload vs launch site





- A slider in the interactive dashboard allows the user to restrict the payloads
- LEFT: payload/success for all payload ranges
- RIGHT: payload/success only for payloads between 4000 kg and 7000 kg

Section 5

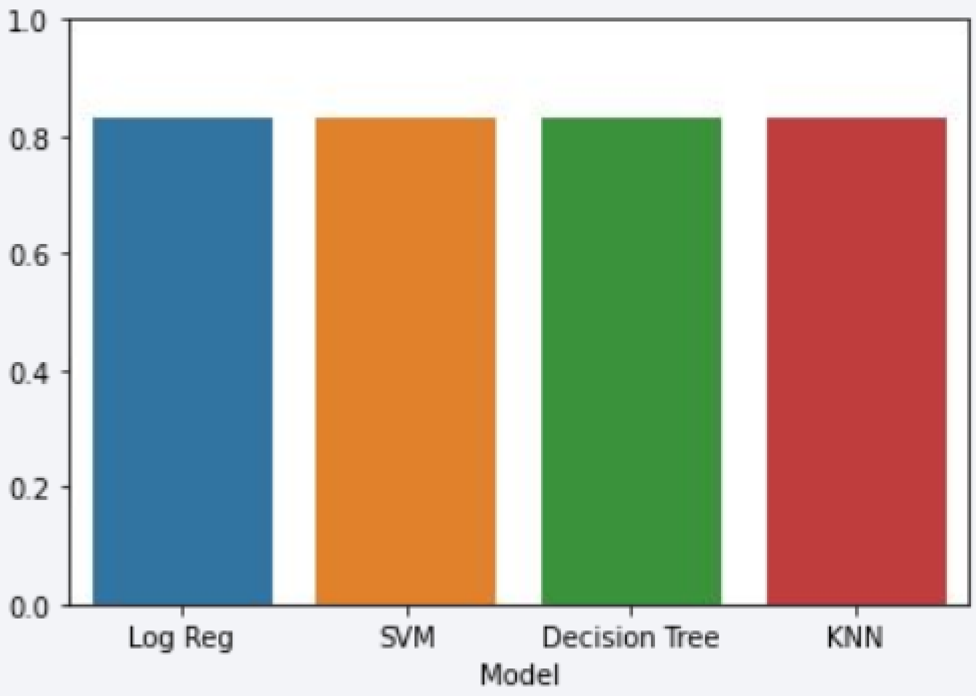
Predictive Analysis (Classification)

Classification Accuracy

For models were created, and the accuracy on the test data was computed:

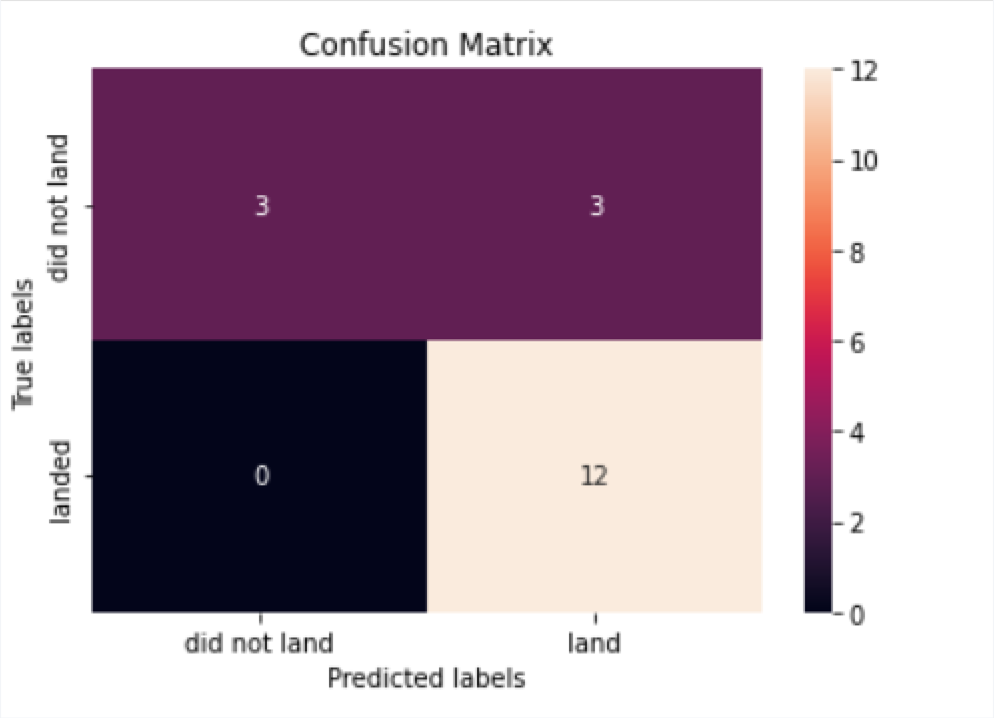
- LogReg **83.3%**
- SVM **83.3%**
- Tree **83.3%**
- KNN **83.3%**

On the validation set the scores were different, but this does not play any role.



Confusion Matrix

- All models had **the same confusion matrix** on the test data!
- As the dataset was extremely small, the test-set was super small (only 18 rows)
- The only problem were false positives, but only three cases



Conclusions

- **We simply did not have enough data**
  Doing ML on 90 records probably creates more confusion than real insight.
- Since SpaceX had already achieved a success rate for Booster recovery of over 80% in 2017, our ML models and predictions (83% accuracy!) are not impressive

But if we take the data seriously, we could provide some hints:

- KSC site is very successful
- Launches with a payload above 7000kg might be less risky
- Some orbits are better than others, this should be investigated

Thank you!