IBM Applied Data Science Space-X

Miklos Csepi 04.02.2023

Outline

- Executive summary
- Introduction
- Methodology
- Results
- Conclusions
- Appendix

Executive summary

Executive Summary

Used methodologies to analyze data:

- Data collection about lunch data using SpaceX API
- Data wrangling to find patterns
- Visualizing data to get relations between data
- Used Machine Learning for prediction

Summary of all results

- Collecting valuable data from public sources
- Interactive dashboard
- Crate model for prediction

Introduction

Introduction

Background and context:

In this capstone, it will predict if the Falcon 9 first stage will land successfully. It can determine if the first stage will land, it can determine the cost of a launch. This information can be used if Space Y company wants to bid against SpaceX for a rocket launch.

Deliverables

To estimate, by what factors can have successful landings

Where is the best place to make space launch

Methodology

Collecting the data

Collecting the data

- use the RESTful API to extract information using identification numbers in the launch data
- Request and parse SpaceX launch data using GET request

ightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude
1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.743129
2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.743129
4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.743129
5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin3C	167.743129
6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366

Collecting the data

- Filter the dataframe to only include Falcon 9 launches
- Data wrangling
- Dealing with Missing Values

```
# Calculate the mean value of PayloadMass column
avg_payload_mass = data_falcon9["PayloadMass"].astype("float").mean(axis=0)
# Replace the np.nan values with its mean value
data_falcon9["PayloadMass"].replace(np.nan, avg_payload_mass, inplace = True)
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Collecting the data

Data wrangling

- Loading SpaceX dataset
- Identify and calculate the percentage of the missing values in each attribute
- Identify which columns are numerical and categorical
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome per orbit type

Collecting the data

Data wrangling

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

- Request the Falcon9 Launch Wiki page from its URL
- Extract all column/variable names from the HTML table header
- Create a data frame by parsing the launch HTML tables

Collecting the data

Data wrangling

- Create a landing outcome label from Outcome column

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
```

```
#df['Class']=landing_class
df['Class'] = df['Outcome'].apply(lambda landing_class: 0 if landing_class in bad_outcomes else 1)
df[['Class']].head(8)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Seria
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B000
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B000
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B000
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B100
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B100

Collecting the data with SQL

Understanding SpaceX dataset through executing SQL queries

- Connect to the database : %load_ext sql
- Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

- Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

Collecting the data with SQL

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEXTBL \
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

- List the date when the first successful landing outcome in ground pad was acheived

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad" FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

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

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

Collecting the data with SQL

- List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%';
```

- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Failure%';
```

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

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEXTBL WHERE PAYLOAD_MASS_
```

Collecting the data with SQL

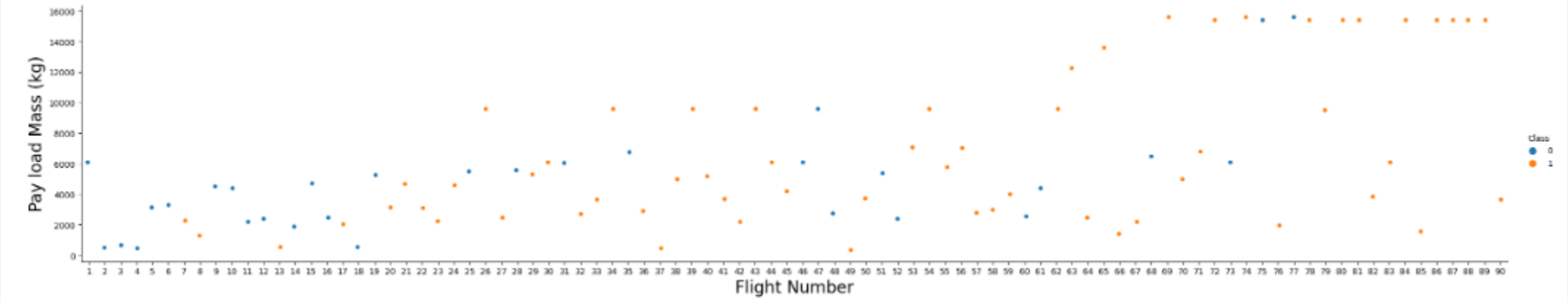
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

Data Visualization

Exploratory Data Analysis

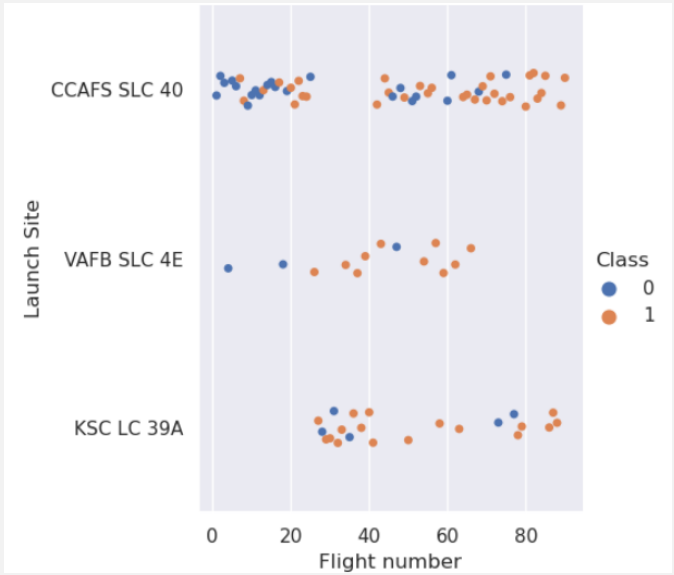
```
sns.catplot(y="PayloadMass", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Pay load Mass (kg)",fontsize=20)
plt.show()
```



Data Visualization

Visualize the relationship between Flight Number and Launch Site

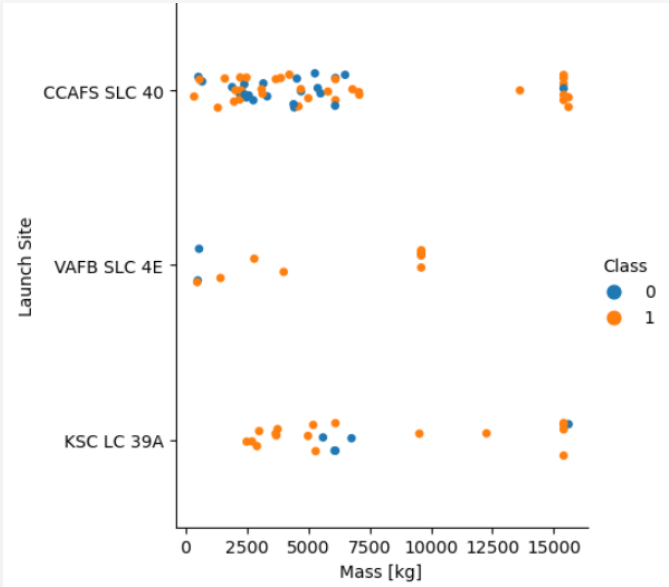
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(data=df, x="FlightNumber", y="LaunchSite", hue="Class", aspect=1)
plt.ylabel("Launch Site")
plt.xlabel("Flight number")
plt.show()
```



Data Visualization

Visualize the relationship between Payload and Launch Site

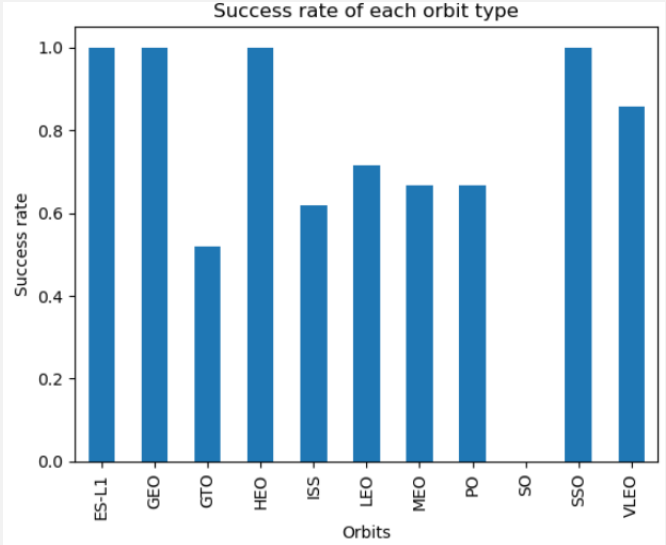
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
ar = sns.catplot(data=df, x="PayloadMass", y="LaunchSite", hue="Class")
ar.set_axis_labels("Mass [kg]", "Launch Site")
ar.set_titles("Payload vs Launch Site")
plt.show()
```



Data Visualization

Visualize the relationship between success rate of each orbit type

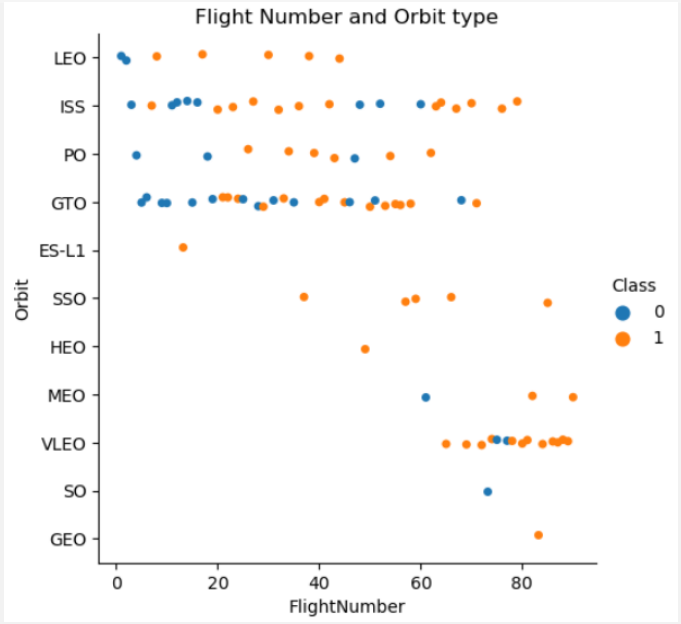
```
# HINT use groupby method on Orbit column and get the mean of Class column
t3=df.groupby("Orbit")["Class"].mean()
ax=t3.plot(kind="bar", color="#1f77b4")
ax.set_xlabel("Orbits")
ax.set_ylabel("Success rate")
plt.title("Success rate of each orbit type")
plt.show()
```



Data Visualization

Visualize the relationship between FlightNumber and Orbit type

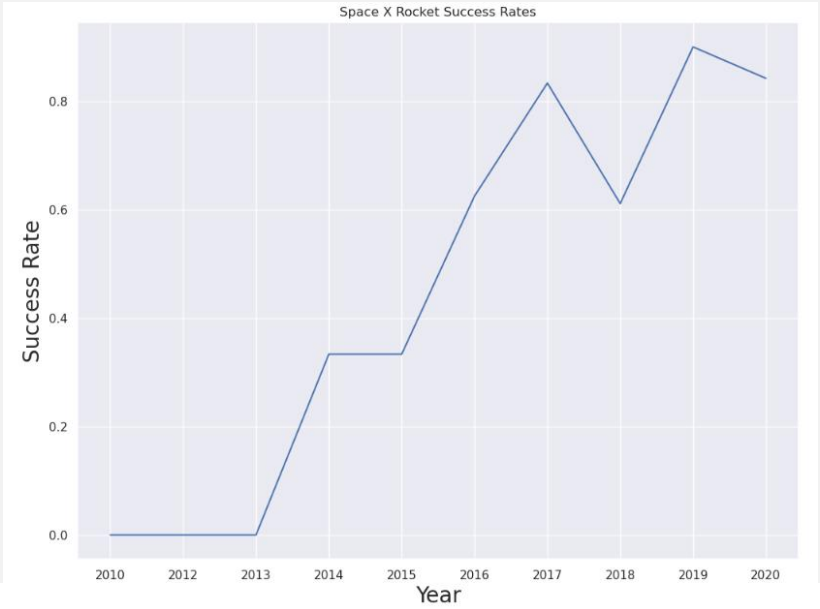
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(data=df, x="FlightNumber", y="Orbit", hue="Class")
plt.title("Flight Number and Orbit type")
plt.xlabel("FlightNumber")
plt.ylabel("Orbit")
plt.show()
```



Data Visualization

Visualize the launch success yearly trend

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
df['year']=Extract_year(df["Date"])
df_groupby_year=df.groupby("year",as_index=False)["Class"].mean()
sns.set(rc={'figure.figsize':(12,9)})
sns.lineplot(data=df_groupby_year, x="year", y="Class" )
plt.xlabel("Year",fontsize=20)
plt.title('Space X Rocket Success Rates')
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



Features Engineering

Select the features that will be used in success prediction in the future module.

```
features = df[['FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flights', 'GridFins', 'Reused', 'Legs', 'LandingPad', 'Block', 'ReusedCount', 'S
features.head()
```

	FlightNumber	PayloadMass	Orbit	LaunchSite	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
0	1	6104.959412	LEO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0003
1	2	525.000000	LEO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0005
2	3	677.000000	ISS	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0007
3	4	500.000000	PO	VAFB SLC 4E	1	False	False	False	NaN	1.0	0	B1003
4	5	3170.000000	GTO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B1004

- Create dummy variables to categorical column

```
# HINT: Use get_dummies() function on the categorical columns
features_one_hot = pd.get_dummies(features, columns=['Orbit', 'LaunchSite', 'LandingPad', 'Serial'])
features_one_hot.head()
```

FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	...	Serial_B1048	Serial_B1049	Serial_B1050	Serial_B1051	Serial_B1054
1	6104.959412	1	False	False	False	1.0	0	0	0	...	0	0	0	0	0
2	525.000000	1	False	False	False	1.0	0	0	0	...	0	0	0	0	0
3	677.000000	1	False	False	False	1.0	0	0	0	...	0	0	0	0	0
4	500.000000	1	False	False	False	1.0	0	0	0	...	0	0	0	0	0
5	3170.000000	1	False	False	False	1.0	0	0	0	...	0	0	0	0	0

Features Engineering

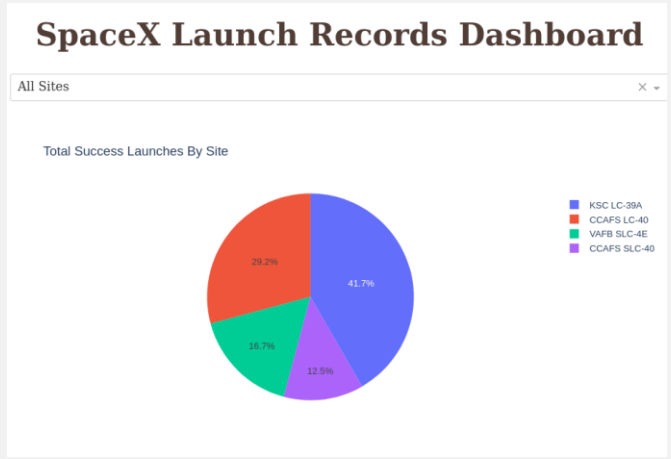
- Cast all numeric columns to float64

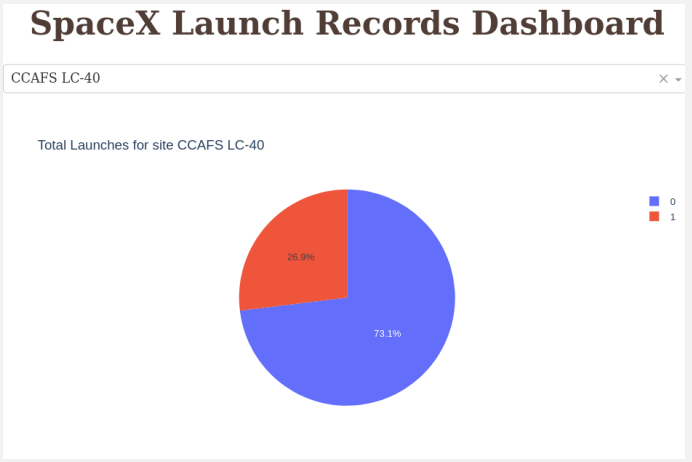
```
# HINT: use astype function
features_one_hot.astype(float)
```

	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	...	Serial_B1048	Serial_B1049	Serial_B1050	Serial_B1051	Serial_B10
0	1.0	6104.959412	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
1	2.0	525.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
2	3.0	677.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
3	4.0	500.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
4	5.0	3170.000000	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
85	86.0	15400.000000	2.0	1.0	1.0	1.0	5.0	2.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
86	87.0	15400.000000	3.0	1.0	1.0	1.0	5.0	2.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
87	88.0	15400.000000	6.0	1.0	1.0	1.0	5.0	5.0	0.0	0.0	...	0.0	0.0	0.0	1.0	
88	89.0	15400.000000	3.0	1.0	1.0	1.0	5.0	2.0	0.0	0.0	...	0.0	0.0	0.0	0.0	
89	90.0	3681.000000	1.0	1.0	0.0	1.0	5.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	

```
features_one_hot.to_csv('dataset_part_3.csv', index=False)
```

Dashboard with Plotly Dash





Results

Results

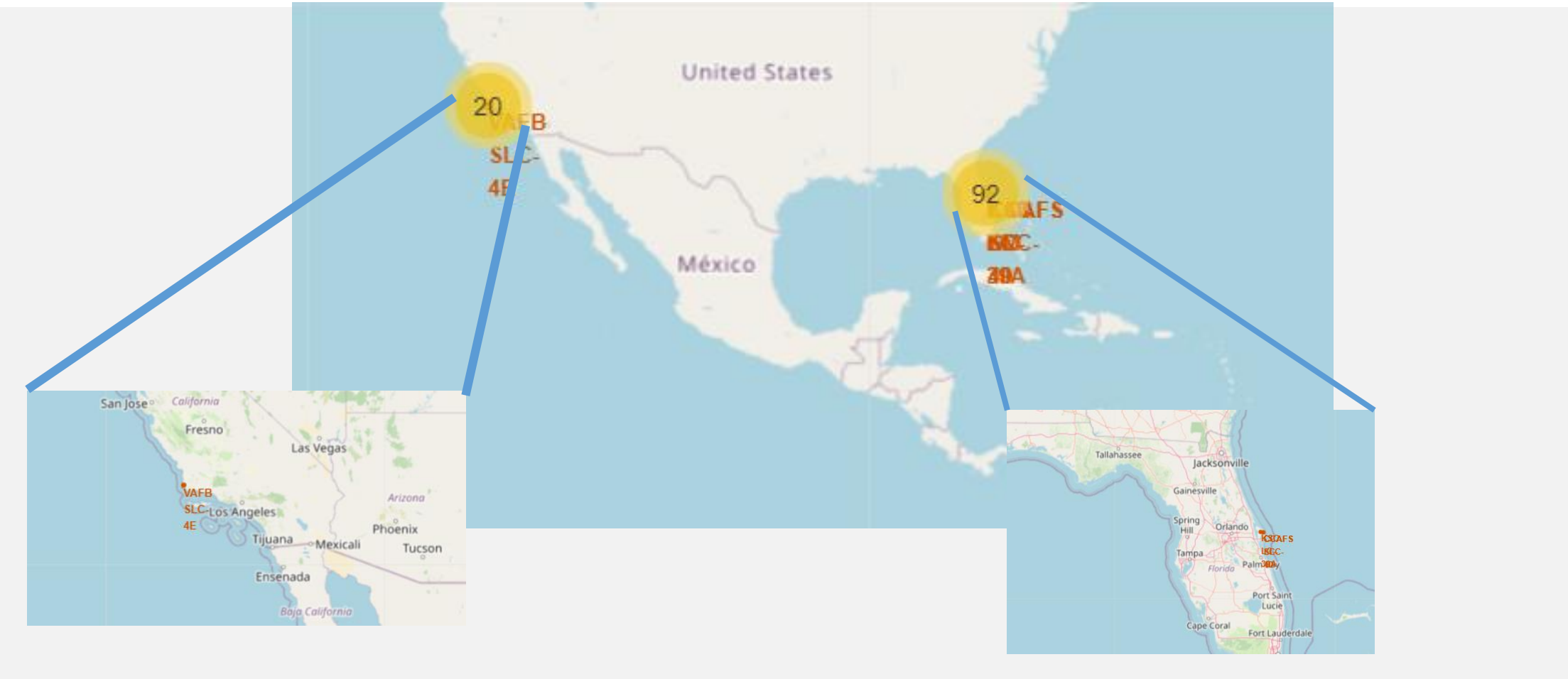
Results out of data analysis

- Different launch sites have different success rate:
  - CCAS LC-40 – 60%
  - KSC LC-39A & VAFB SLC 4E – 77%
- Success rate become over 60% from 2016
- Success rate 100%: ES-L1, GEO, HEO, SSO
- VLEO orbit type has the most success flights, over 80
- Most payloads with successful launches are from CCAFS SLC 40 & KSC LC39A

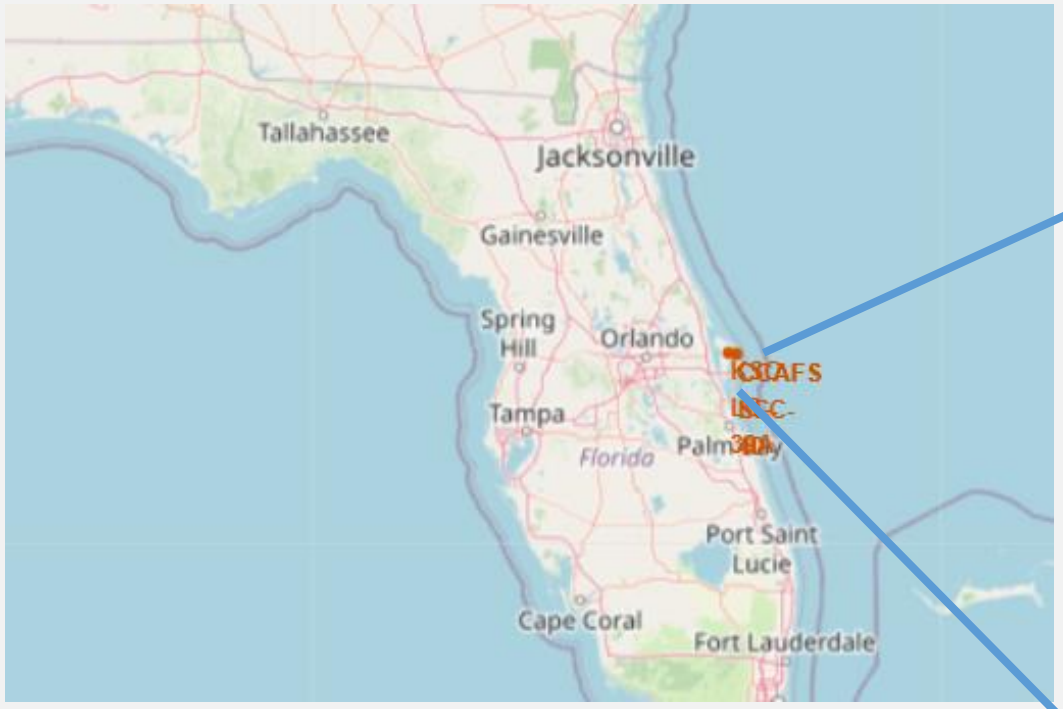
Results

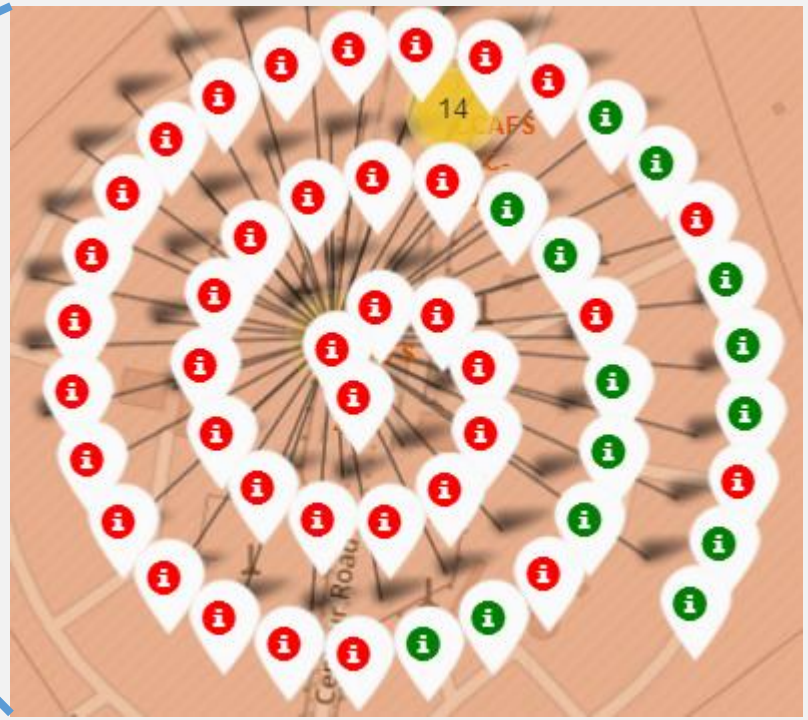
- AVG Payloads,2.928 kg by booster version F9
- 99 Success launches have been made
- Total Payload Mass by NASA: 45596 kg
- First successful landing outcome in Ground Pad: 22-12-2015

Maps


United States
México
20
92
San Jose
California
Fresno
Las Vegas
VAFB
SLC-
4E
Los Angeles
Arizona
Phoenix
Tijuana
Mexicali
Tucson
Ensenada
Baja California
Tallahassee
Jacksonville
Gainesville
Spring Hill
Orlando
Tampa
Florida
Port Saint Lucie
Cape Coral
Fort Lauderdale

Maps


Tallahassee
Jacksonville
Gainesville
Spring Hill
Orlando
Tampa
Florida
Port Saint Lucie
Cape Coral
Fort Lauderdale

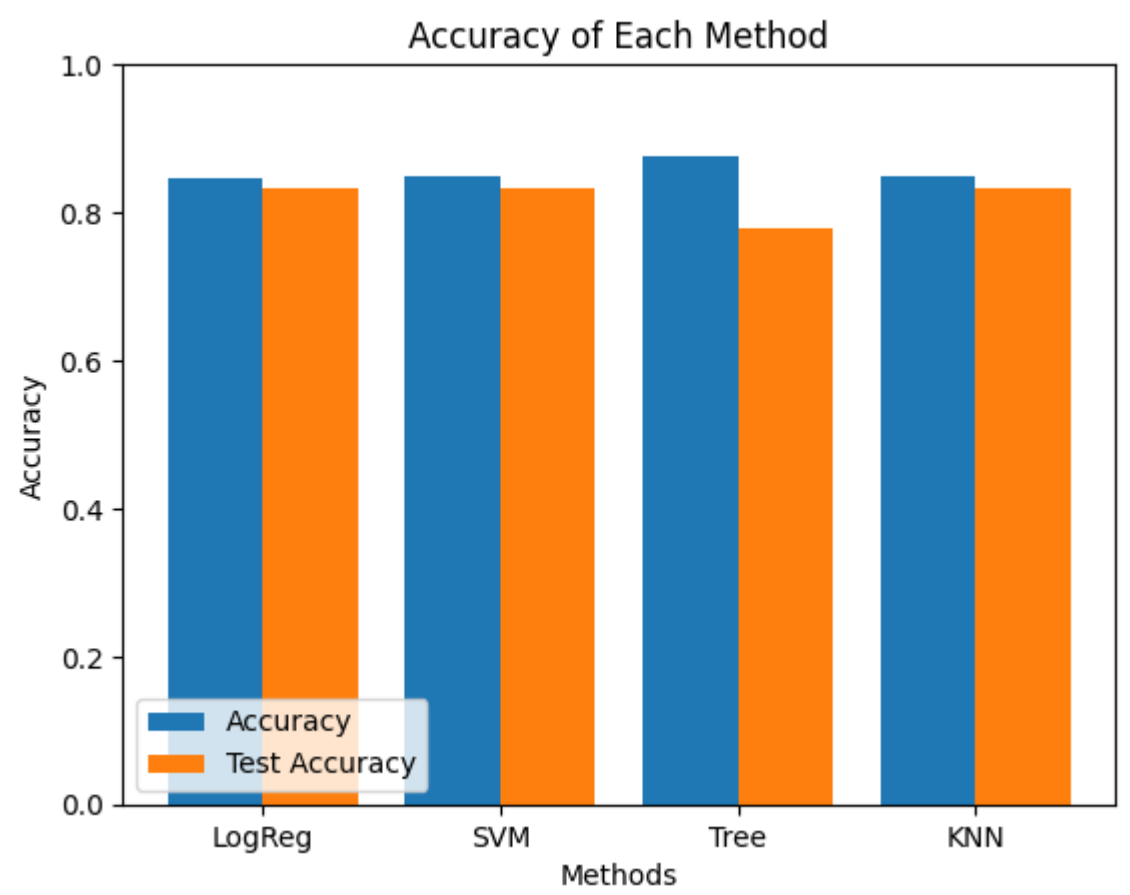

14

Prediction

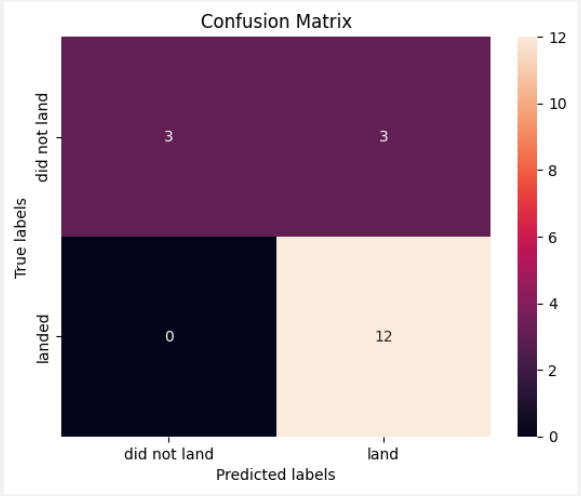
Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.87679	0.77778
KNN	0.84821	0.83333

Four classifications models were tested

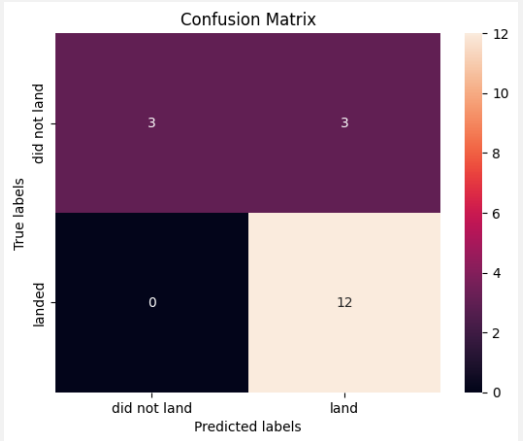
Tree model has the highest accuracy however, its test accuracy was the lowest



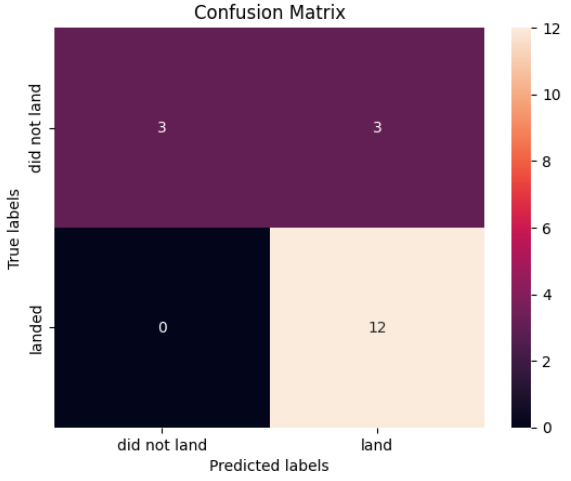
Prediction



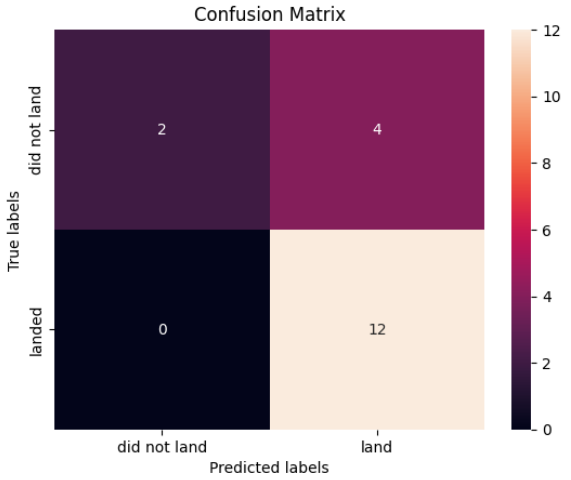
Logreg



KNN



SVM



Tree

Conclusions

Conclusions

- SpaceX success rates are increasing over the years
- Best launch site is KSC LC-39A
- Launches become risky with Payloads above 7000 kg
- To predict successful landings Decision Tree Classifier worth being used

Appendix

- Applied Data Science Capstone: https://www.coursera.org/learn/applied-data-science-capstone
- GitHub-Link: https://github.com/mcsepi/IBM-Applied-Data-Schience-Capstone

Thank You!