SpaceX Falcon 9

Hani Ahmed 28/12/2022

Executive summary

- if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. In this project, we will collect and make sure the data is in the correct format from an API. The following is an example of a successful and launch
- We will EDA the data and build the best model to answer our questions

Introduction

- we 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.

Required data collection and wrangling methodology

- **Our Objectives**

- In this stage, we will make a get request to the SpaceX API. we will also do some basic data wrangling and formating.

- Request to the SpaceX API
- Clean the requested data

[24]:

	FlightNumber	PayloadMass	Flights	Block	ReusedCount	Longitude	Latitude
count	94.000000	88.000000	94.000000	90.000000	94.000000	94.000000	94.000000
mean	54.202128	5919.165341	1.755319	3.500000	3.053191	-75.553302	28.581782
std	30.589048	4909.689575	1.197544	1.595288	4.153938	53.391880	4.639981
min	1.000000	20.000000	1.000000	1.000000	0.000000	-120.610829	9.047721
25%	28.250000	2406.250000	1.000000	2.000000	0.000000	-80.603956	28.561857
50%	52.500000	4414.000000	1.000000	4.000000	1.000000	-80.577366	28.561857
75%	81.500000	9543.750000	2.000000	5.000000	4.000000	-80.577366	28.608058
max	106.000000	15600.000000	6.000000	5.000000	13.000000	167.743129	34.632093

Data Wrangling

We can see below that some of the rows are missing values in our dataset.

[64]:

```
data_falcon9.isnull().sum()
```

[64]:

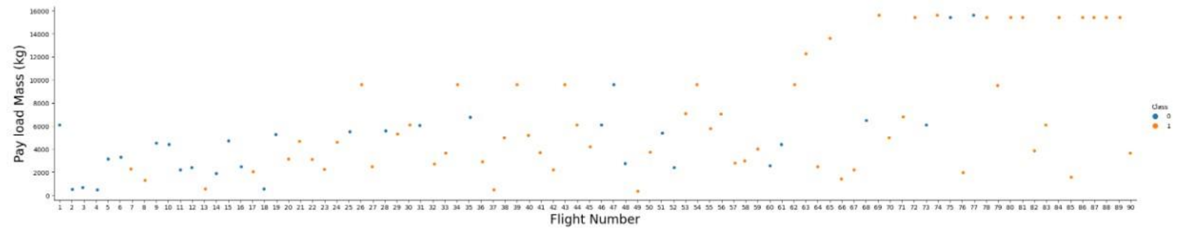
```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        0
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

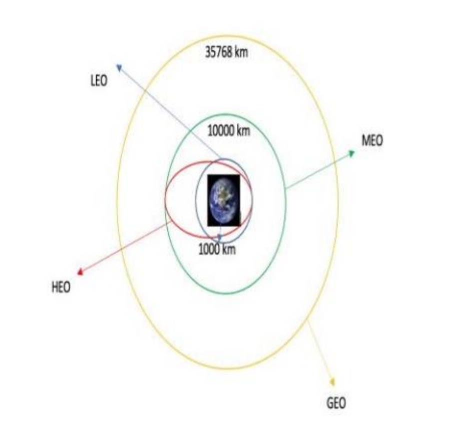
EDA and interactive visual analytics methodology

- **Our Objectives**
- Perform exploratory Data Analysis and Feature Engineering using Pandas and Matplotlib
- Exploratory Data Analysis
- Preparing Data Feature Engineering

[2]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0





TASK 2: Calculate the number and occurrence of each orbit

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

```
# Apply value_counts on Orbit column
df.Orbit.value_counts()
```

```
GTO       27
ISS       21
VLEO      14
PO         9
LEO        7
SSO        5
MEO        3
ES-L1      1
HEO        1
SO         1
GEO        1
Name: Orbit, dtype: int64
```

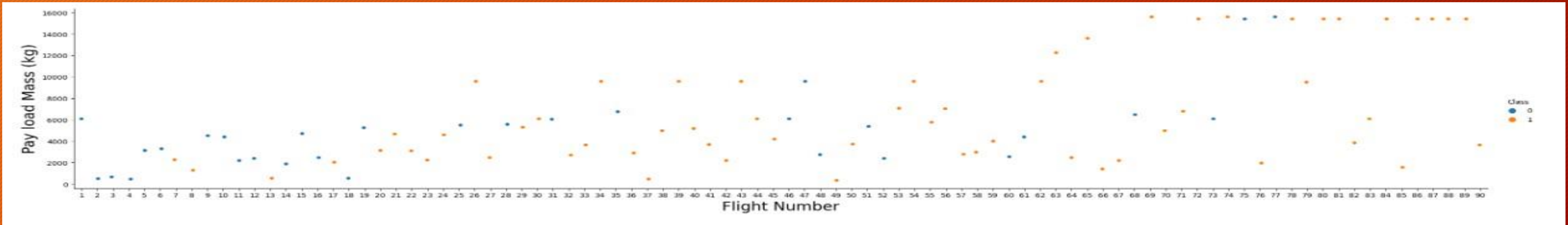
predictive analysis methodology

- **Objectives**
- Perform exploratory Data Analysis and determine Training Labels
- 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

EDA with visualization results

[2]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0



[24]:

	FlightNumber	PayloadMass	Flights	Block	ReusedCount	Longitude	Latitude
count	94.000000	88.000000	94.000000	90.000000	94.000000	94.000000	94.000000
mean	54.202128	5919.165341	1.755319	3.500000	3.053191	-75.553302	28.581782
std	30.589048	4909.689575	1.197544	1.595288	4.153938	53.391880	4.639981
min	1.000000	20.000000	1.000000	1.000000	0.000000	-120.610829	9.047721
25%	28.250000	2406.250000	1.000000	2.000000	0.000000	-80.603956	28.561857
50%	52.500000	4414.000000	1.000000	4.000000	1.000000	-80.577366	28.561857
75%	81.500000	9543.750000	2.000000	5.000000	4.000000	-80.577366	28.608058
max	106.000000	15600.000000	6.000000	5.000000	13.000000	167.743129	34.632093

EDA with SQL results

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

[13]:
```
data[data.Launch_Site.str.startswith('CCA')].head()
```

[13]:

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
0	04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

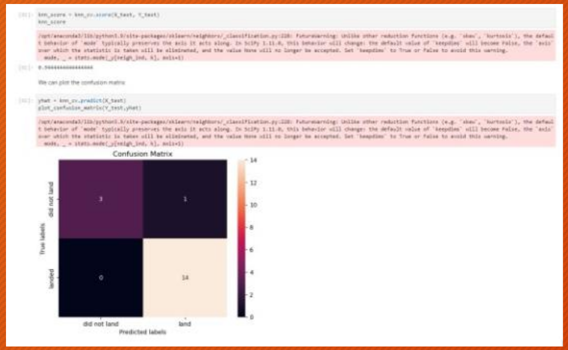
Display the total payload mass carried by boosters launched by NASA (CRS)

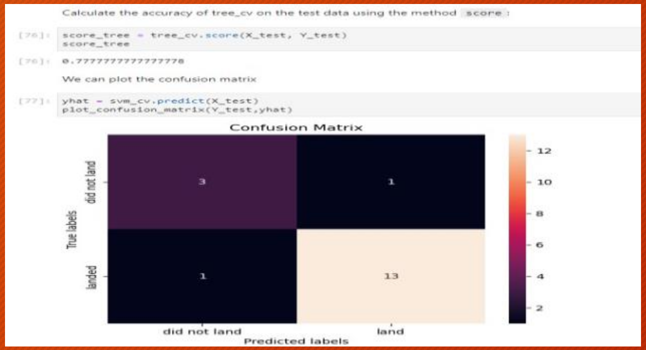
[18]:
```
data[data.Customer=='NASA (CRS)']['PAYLOAD_MASS__KG_'].sum()
```

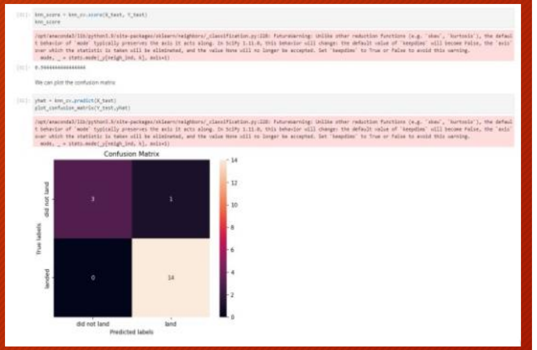
[18]: 45596

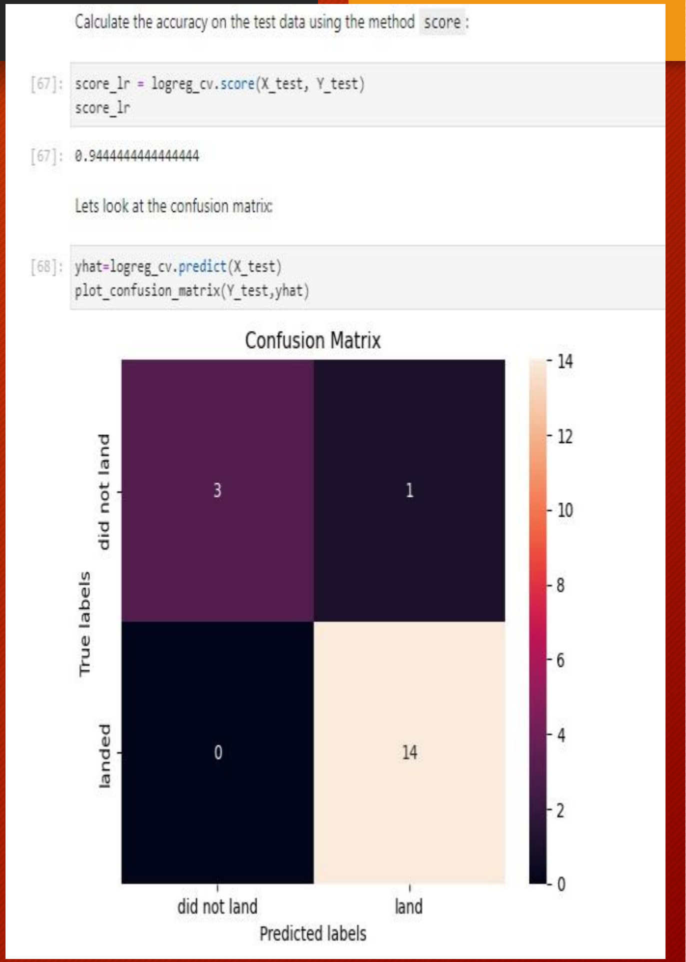
the predictive analysis result

- **Objectives**
- Perform exploratory Data Analysis and determine Training Labels
- 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









the Conclusion

```
[87]: best_model = [score_lr, score_svm, score_tree, knn_score]
      print(np.sort(best_model))
```

```
[0.77777778 0.88888889 0.94444444 0.94444444]
```