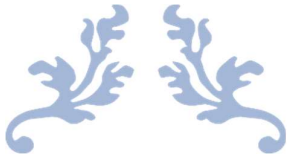


# SPACEX ROCKET PRICE PREDICTION

Capstone project

# Introduction :

The commercial space age is here, companies are making space travel affordable for everyone. Virgin Galactic is providing suborbital spaceflights. Rocket Lab is a small satellite provider. the most successful is SpaceX. SpaceX's accomplishments include:
Sending spacecraft to the International Space Station. Starlink, a satellite internet constellation providing satellite Internet access. Sending manned missions to Space.
One reason SpaceX can do this is the rocket launches are relatively inexpensive.
SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. The purpose of capstone project is to determine the price of each launch. I will be doing this by gathering information about Space X and creating dashboards for your team. And also determining if SpaceX will reuse the first stage.

.

# DATA SECTION :

Reference Data_link : https://en.wikipedia.org/wiki/SpaceX

I used above link to know more about spacex and rest of the data is collected from IBM skills network and used via Coursera plotform and I also observed few vidoes to know how it works

**Data-link : "https://api.spacexdata.com/v4/launches/past"**

Data-link :
https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

we are going to use these diagrams from Forest Katsch, at zlsadesign.com. He is a 3D artist and software engineer. He makes infographics on spaceflight and spacecraft art. He also makes software. The payload is enclosed in the fairings. Stage two, or the second stage, helps bring the payload to orbit, but most of the work is done by the first stage.
The first stage is shown here. This stage does most of the work and is much larger than the second stage.
Here we see the first stage next to a person and several other landmarks. This stage is quite large and expensive. Unlike other rocket providers, SpaceX's Falcon 9 Can recover the first stage.

| | FlightNumber | Date | BoosterVersion | PayloadMass | |
|---|---|---|---|---|---|
| 1 | 1 | 2010-06-04 | Falcon 9 | 6123.547647058824 | |
| 2 | 2 | 2012-05-22 | Falcon 9 | 525.0 | |
| 3 | 3 | 2013-03-01 | Falcon 9 | 677.0 | |
| 4 | 4 | 2013-09-29 | Falcon 9 | 500.0 | |
| 5 | 5 | 2013-12-03 | Falcon 9 | 3170.0 | |
| 6 | 6 | 2014-01-06 | Falcon 9 | 3325.0 | |
| 7 | 7 | 2014-04-18 | Falcon 9 | 2296.0 | |
| 8 | 8 | 2014-07-14 | Falcon 9 | 1316.0 | |
| 9 | 9 | 2014-08-05 | Falcon 9 | 4535.0 | |
| 10 | 10 | 2014-09-07 | Falcon 9 | 4428.0 | |
| 11 | 11 | 2014-09-21 | Falcon 9 | 2216.0 | |
| 12 | 12 | 2015-01-10 | Falcon 9 | 2395.0 | |
| 13 | 13 | 2015-02-11 | Falcon 9 | 570.0 | |
| 14 | 14 | 2015-04-14 | Falcon 9 | 1898.0 | |

# Methodology Section :

Clustering Approach:
To compare the similarities of two cities, we decided to explore neighborhoods, segment them, and group them into clusters

```
2]: df.head(5)
```

2]:

| | FlightNumber | Date | BoosterVersion | PayloadMass | Orbit | LaunchSite | Outcome | Flights | GridFins |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 2010-06-04 | Falcon 9 | 6104.959412 | LEO | CCAFS SLC 40 | None None | 1 | False |
| **1** | 2 | 2012-05-22 | Falcon 9 | 525.000000 | LEO | CCAFS SLC 40 | None None | 1 | False |
| **2** | 3 | 2013-03-01 | Falcon 9 | 677.000000 | ISS | CCAFS SLC 40 | None None | 1 | False |
| **3** | 4 | 2013-09-29 | Falcon 9 | 500.000000 | PO | VAFB SLC 4E | False Ocean | 1 | False |
| **4** | 5 | 2013-12-03 | Falcon 9 | 3170.000000 | GTO | CCAFS SLC 40 | None None | 1 | False |

We can use the following line of code to determine the success rate:

```
3]: df["Class"].mean()
```

```
3]: 0.6666666666666666
```

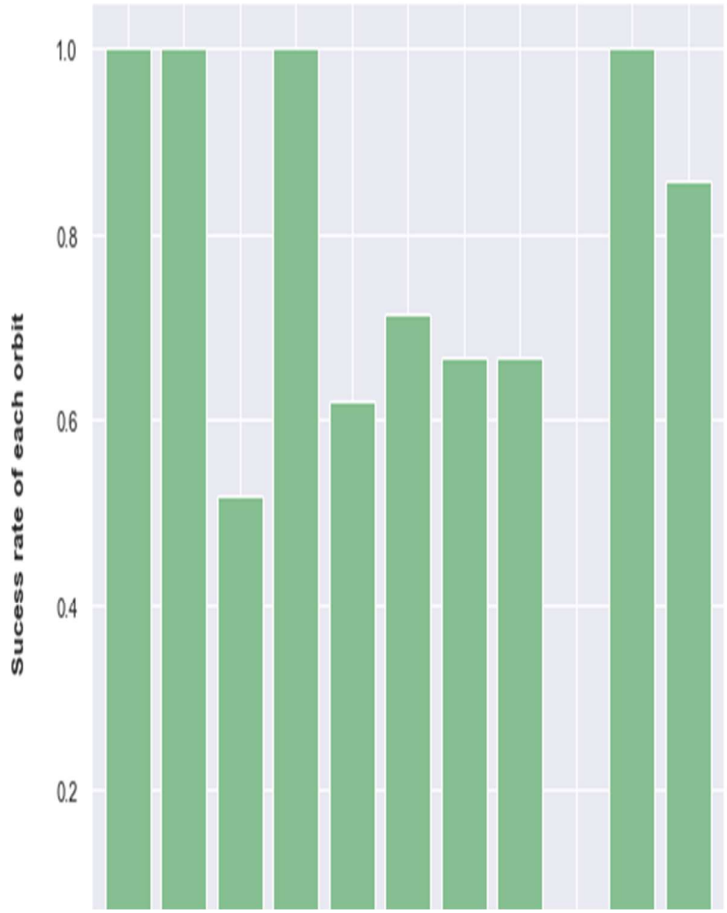
# Results Section :

## TASK 3: Visualize the relationship between success rate of each orbit type

Next, we want to visually check if there are any relationship between success rate and orbit type.

Let's create a `bar chart` for the sucess rate of each orbit

```
3]: # HINT use groupby method on Orbit column and get the mean of Class column
    # Via Matplotlib
    xh = df.groupby('Orbit')['Class'].mean()
    ax = xh.plot(kind='bar', figsize=(8, 7), color='#86bf91', zorder=2, width=0.8)
    ax.set_xlabel("Orbit", labelpad=20, weight='bold', size=12)
    ax.set_ylabel("Sucess rate of each orbit", labelpad=20, weight='bold', size=12);
```



Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the c

[20]:
```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count"
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.datab
Done.
```

[20]:

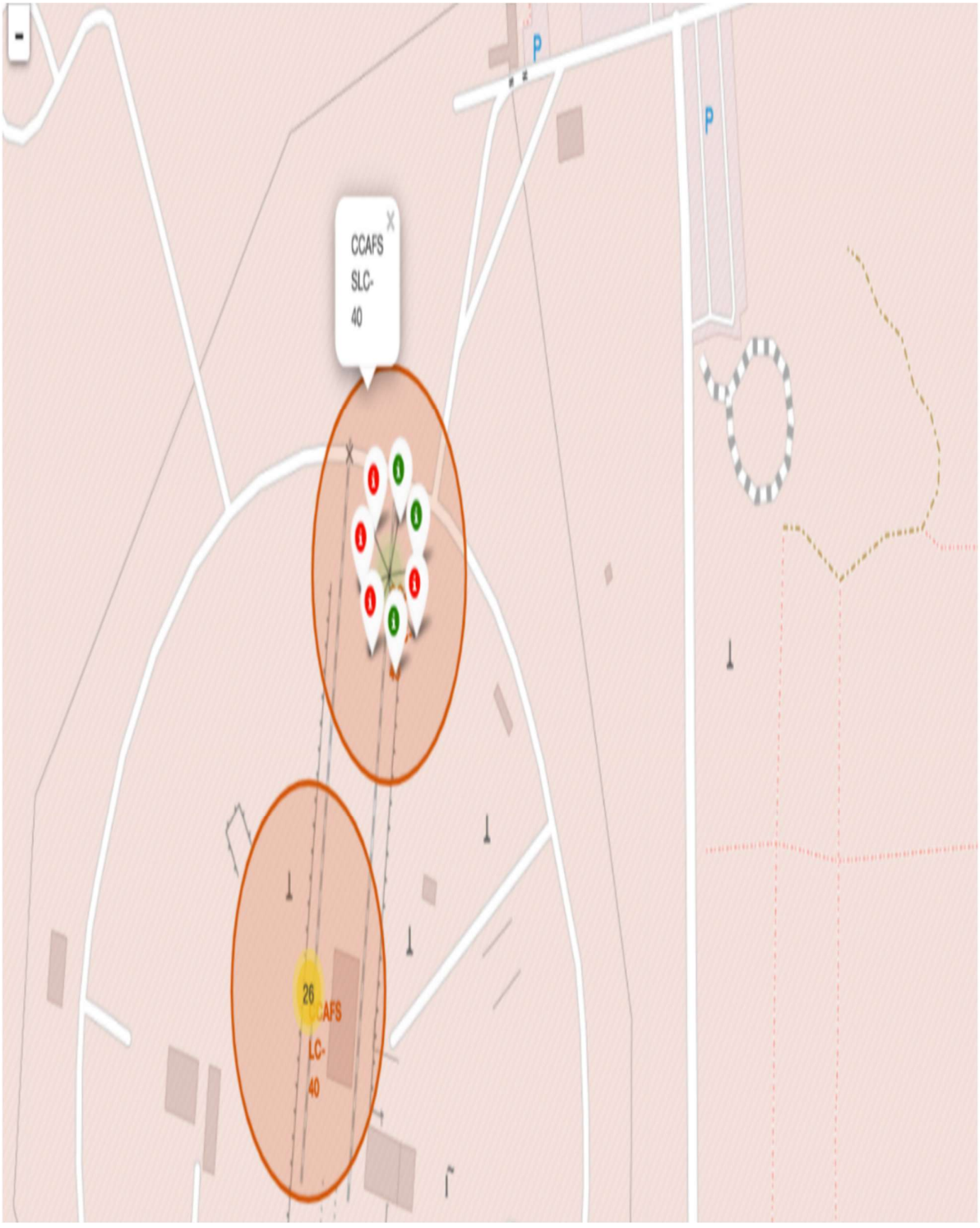
| Landing Outcome | Total Count |
|---|---|
| No attempt | 10 |
| Failure (drone ship) | 5 |
| Success (drone ship) | 5 |
| Controlled (ocean) | 3 |
| Success (ground pad) | 3 |
| Failure (parachute) | 2 |
| Uncontrolled (ocean) | 2 |
| Precluded (drone ship) | 1 |

[21]:
```
%sql SELECT COUNT(LANDING__OUTCOME) AS "Rank success count between 2010-06-04 and 2017-03-2
WHERE LANDING__OUTCOME LIKE '%Success%' AND DATE > '2010-06-04' AND DATE < '2017-03-20' ;
```

```
 * ibm_db_sa://zpw86771:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.datab
Done.
```

[21]:

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


CCAFS
SLC-
40
P
P
26
AFS
LC-
40

| | |
|---|---|
| **KNN** | 0.848214 |
| **Decision Tree** | 0.901786 |
| **Logistic Regression** | 0.846429 |
| **SVM** | 0.848214 |

[33]:
```
algo_df.sort_values(['Accuracy'], inplace=True)
```

[34]:
```
algo_df.head()
```

[34]:

| | Accuracy |
|---|---|
| **Logistic Regression** | 0.846429 |
| **SVM** | 0.848214 |
| **KNN** | 0.848214 |
| **Decision Tree** | 0.901786 |

[35]:
```
algo_df = algo_df.reset_index()
algo_df.head()
```

[35]:

| | index | Accuracy |
|---|---|---|
| **0** | Logistic Regression | 0.846429 |
| **1** | SVM | 0.848214 |
| **2** | KNN | 0.848214 |
| **3** | Decision Tree | 0.901786 |

[36]:
```
algo_df.rename(columns = {'index': 'Algorithm'}, inplace = True)
algo_df.head()
```

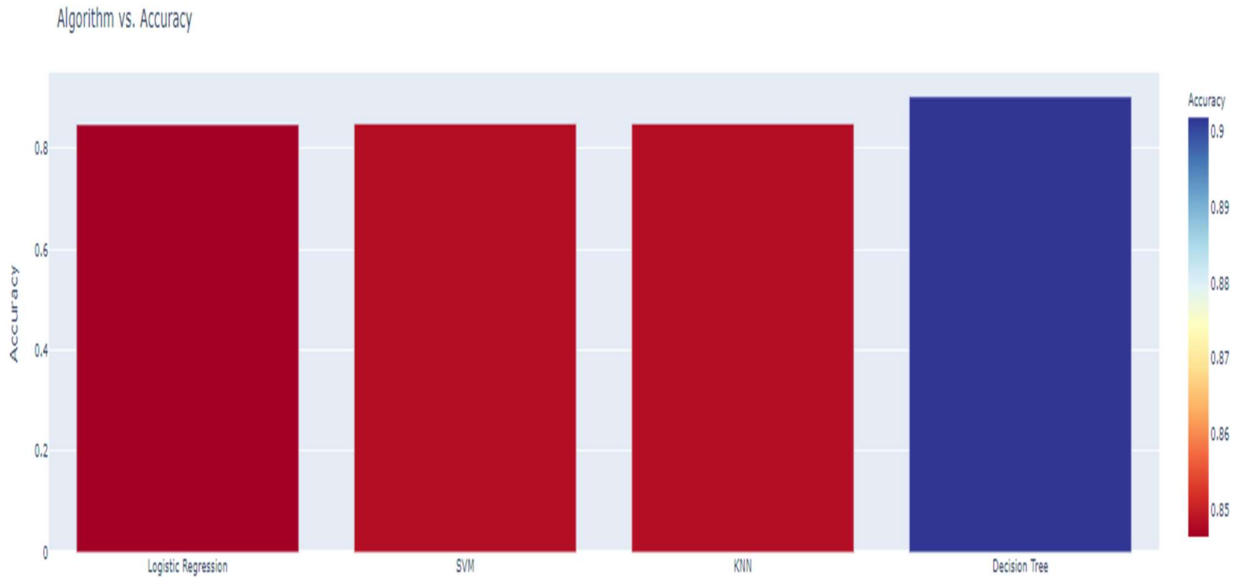
[36]:

| | Algorithm | Accuracy |
|---|---|---|
| **0** | Logistic Regression | 0.846429 |
| **1** | SVM | 0.848214 |
| **2** | KNN | 0.848214 |
| **3** | Decision Tree | 0.901786 |

| | Algorithm | Accuracy |
|---|---|---|
| 0 | Logistic Regression | 0.846429 |
| 1 | SVM | 0.848214 |
| 2 | KNN | 0.848214 |
| 3 | Decision Tree | 0.901786 |

```
import plotly.express as px
import plotly.graph_objects as go
```

```
fig = px.bar(algo_df, x='Algorithm', y='Accuracy', hover_data=['Algorithm', 'Accuracy'], color='Accuracy', color_continuous_scale='rdylbu')
fig.update_layout(title='Algorithm vs. Accuracy', xaxis_title='Algorithm', yaxis_title='Accuracy' )
fig.show()
```



# Discussion Section

Problem Which Tried to Solve:

.The purpose of capstone project is to determine the price of each launch. And also to find if we can use the first satge repeatatively and share findings as in last slides

Here we solved the problem and shared the result in pictorial format as in previous slides

# Conclusion :

The commercial space age is here, companies are making space travel affordable for everyone . the most successful is SpaceX. SpaceX's accomplishments include: Sending spacecraft to the International Space Station and we did this capstone project to see

1. To see if stage 1 can be performed numerous times
2. To find out price for the work and success rate and few more analytical stuff

Thus I conclude accuracy in the pictorial format in the pics beside

Libraries Which are Used to Develop the Project:

Pandas: For creating and manipulating dataframes.

Folium: Python visualization library would be used to visualize the neighborhoods cluster distribution of using interactive leaflet map.

Scikit Learn: For importing k-means clustering.

JSON: Library to handle JSON files.

XML: To separate data from presentation and XML stores data in plain text format.

Geocoder: To retrieve Location Data.

Beautiful Soup and Requests: To scrap and library to handle http requests.

Matplotlib: Python Plotting Module.

----------The End----------