



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Macedo, Glener Diniz
14 de dezembro de 2024



Outline

- Resumo Executivo
- Introdução
- Metodologia
- Resultados
- Conclusão
- Apêndice



Executive Summary

Esse estudo visa identificar os fatores para um pouso de foguetes de modo que sejam bem-sucedido. Para fazer essa viabilidade, foram usados os seguintes metodos:

- Coleta de dados usando a API REST da SpaceX e técnicas de web scraping;
- Disputa de dados para criação de variáveis de resultado de sucesso e de falha;
- Exploração com técnicas de visualização de dados, considerando os seguintes fatores:
 - carga útil, local de lançamento, número de voo e tendência annual
- Analise de dados com SQL, calculando as seguintes estatísticas:
 - carga útil total, intervalo de carga útil para lançamentos bem-sucedidos e total de resultados bem-sucedidos e malsucedidos;
- Exploração das taxas de sucesso do local de lançamento e proximidade de marcadores geográficos;
- Visualização dos locais de lançamento com mais sucesso e intervalos de carga útil bem-sucedidos;
- Construção de modelos para prever resultados de pouso usando regressão logística, máquina de vetores de suporte (SVM), árvore de decisão e K-vizinho mais próximo (KNN).

Resultados: Análise de dados exploratórios:

- O sucesso do lançamento melhorou ao longo do tempo;
- O KSC LC-39A tem a maior taxa de sucesso entre os pontos de pouso;
- As órbitas ES-L1, GEO, HEO e SSO têm 100% de taxa de sucesso.

Resumo Executivo Visualização/Análise: A maioria dos locais de lançamento foram realizados próximo da linha do equador, e todos estão perto do litoral.

Análise Preditiva: Todos os modelos tiveram desempenho semelhante no conjunto de teste. O modelo de árvore de decisão teve mais sucesso.



Introduction

Ser líder na indústria espacial, se esforça para tornar as viagens espaciais acessíveis para todos. Com realizações que incluem enviar espaçonaves para a estação espacial internacional, lançar uma seleção de satélites que fornece acesso à internet e enviar missões tripuladas ao espaço. Estamos falando da SpaceX, empresa que visa fazer isso porque os lançamentos de foguetes são relativamente baratos (US\$ 62 milhões por lançamento) devido à sua nova reutilização do primeiro estágio do seu foguete Falcon 9.

Outros provedores, que não conseguem reutilizar o primeiro estágio, custam mais de US\$ 165 milhões cada.

Ao determinar se o primeiro estágio pousará, podemos determinar o valor de custo do lançamento.

Para fazer isso, podemos usar dados públicos e modelos de aprendizado de máquina para prever se a SpaceX – ou uma empresa concorrente – pode reutilizar o primeiro estágio.



Problemas para os quais você deseja encontrar respostas:

- Como a massa da carga útil, o local de lançamento, o número de voos e as órbitas afetam o sucesso do pouso do primeiro estágio;
- Taxa de pousos bem-sucedidos ao longo do tempo;
- Melhor modelo preditivo para pouso bem-sucedido (classificação binária).



Section 1

Methodology

Methodology

Etapas

- Coletar dados usando a API REST do SpaceX e técnicas de web scraping
- Disputar dados – filtrando os dados, manipulando valores ausentes e aplicando

uma codificação quente – para preparar os dados para análise e modelagem

- Explorar dados via EDA com SQL e técnicas de visualização de dados
- Visualizar os dados usando Folium e Plotly Dash
- Construir modelos para prever resultados de pouso usando modelos de classificação. Ajustar e avaliar modelos para encontrar o melhor modelo e parâmetros

Data Collection

Etapas

- Solicitar dados de lançamento de foguete da API SpaceX;
- Decodifique a resposta usando `.json()` e converta para um dataframe usando `.json_normalize()`;
- Solicitar informações sobre os lançamentos da API SpaceX usando funções personalizadas;
- Criar dicionário a partir dos dados;
- Criar dataframe a partir do dicionário;
- Filtrar dataframe para conter apenas lançamentos do Falcon 9;
- Substituir valores ausentes de Payload Mass por `.mean()` calculado;
- Exportar dados para arquivo csv.



Data Collection – SpaceX API

- A SpaceX ofereceu uma API pública de onde os dados puderam ser obtidos e armazenados e processados;
- Esta API foi usada de acordo com o fluxograma ao lado e então os dados foram persistidos.



<https://github.com/gdmacedo/applied-data-science-capstone/blob/main/1.3.1%20-%20Complete%20the%20Data%20Collection%20API%20Lab.ipynb>

Data Collection - Scraping

- Dados dos lançamentos da SpaceX também podem ser obtidos da Wikipedia;
- Os dados são baixados da Wikipedia de acordo com o fluxograma e então persistidos.

Request API and parse the
SpaceX launch data



Filter data to only include Falcon 9 launches



Deal with Missing Values

<https://github.com/gdmacedo/applied-data-science-capstone/blob/main/1.3.2%20-%20Data%20Collection%20with%20Web%20Scraping.ipynb>

Data Wrangling

Realizaremos algumas Análises Exploratórias de Dados (EDA) para encontrar alguns padrões nos dados e determinar qual seria o rótulo para treinar modelos supervisionados.

Com vários casos diferentes em que o propulsor não pousou com sucesso, sendo esses RTLS, ASDS True ou False.

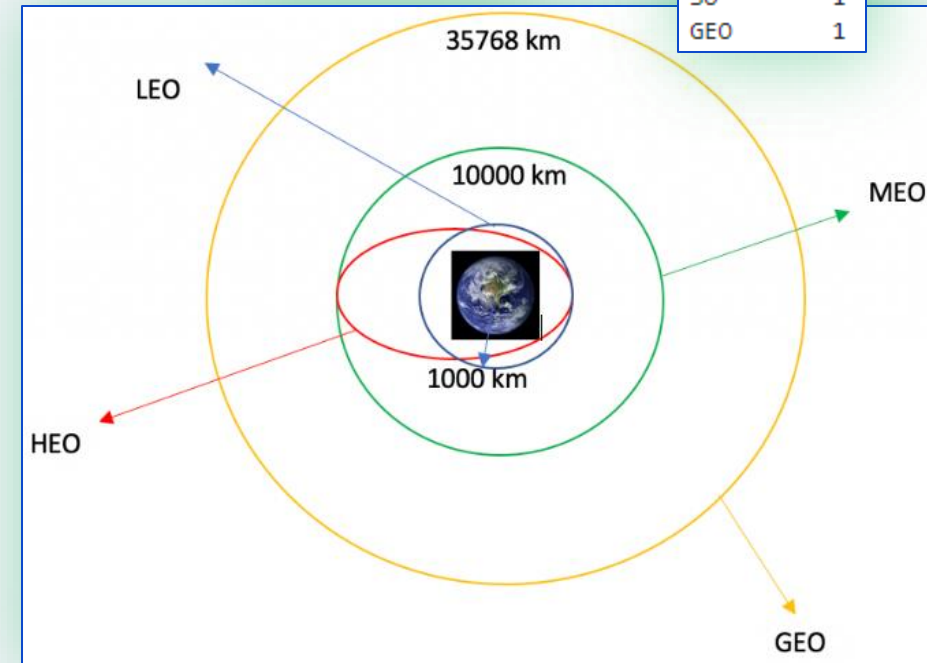
True ou False Ocean - Resultado da missão foi pousado com ou sem sucesso em uma região específica do oceano,

Convertendo principalmente os resultados em rótulos de treinamento, com 1 significa que o propulsor pousou com sucesso e 0 significa que não teve sucesso.

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
df['Outcome'].value_counts()
```

Outcome	
True ASDS	41
None None	19
True RTLS	14
False ASDS	6
True Ocean	5
False Ocean	2
None ASDS	2
False RTLS	1

Orbit	
GTO	27
ISS	21
VLEO	14
PO	9
LEO	7
SSO	5
MEO	3
HEO	1
ES-L1	1
SO	1
GEO	1



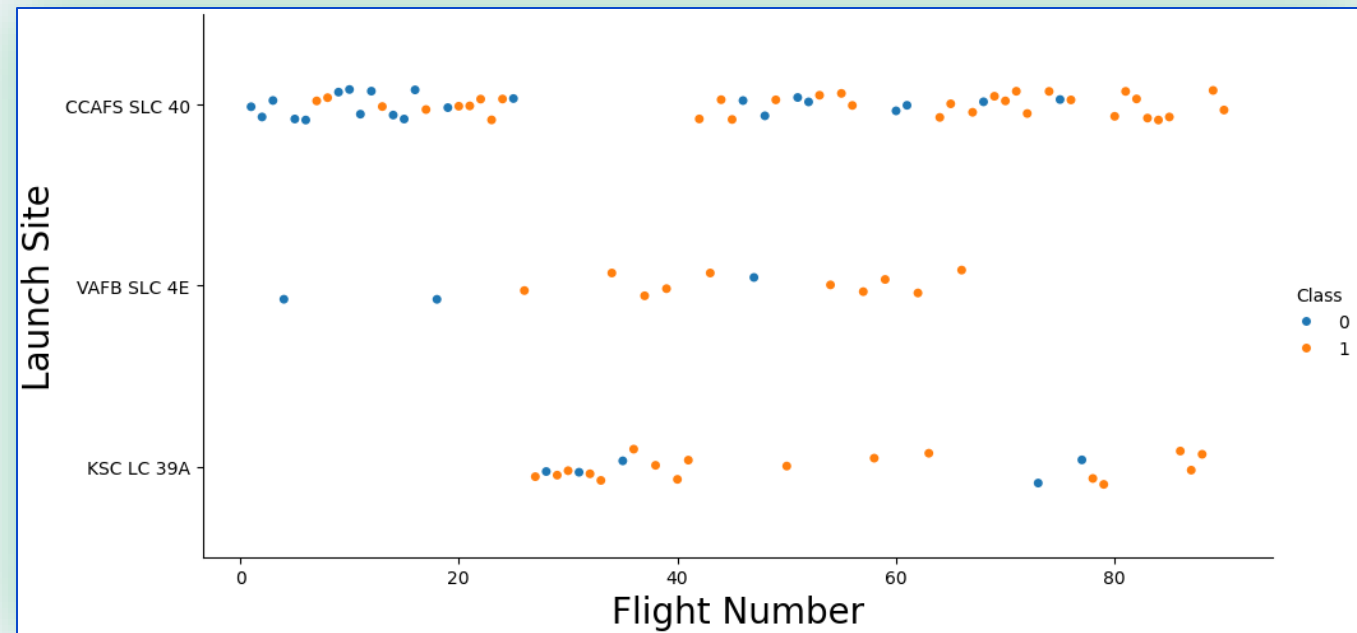
<https://github.com/gdmacedo/applied-data-science-capstone/blob/main/1.4.2%20-%20labs-jupyter-spacex-Data%20wrangling.ipynb>

EDA with Data Visualization

Podemos prever se o primeiro estágio do Falcon 9 pousará com sucesso, executando a análise exploratória de dados e engenharia de recursos.

```
# 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(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect=2)  
plt.xlabel("Flight Number", fontsize=20)  
plt.ylabel("Launch Site", fontsize=20)  
plt.show()
```

A SpaceX anuncia lançamentos dos foguetes em seu site com um custo de 62 milhões de dólares; outros provedores custam mais de 165 milhões de dólares cada, grande parte da economia se deve ao fato de que a SpaceX pode reutilizar o primeiro estágio.



EDA with SQL

Foram realizadas as seguintes consultas em SQL:

- Nomes dos locais de lançamento exclusivos na missão espacial;
- Os 5 principais locais de lançamento cujo nome começa com a string 'CCA';
- Massa total de carga útil transportada pelos propulsores lançados pela NASA (CRS);
- Massa média de carga útil transportada pela versão F9 v1.1 do propulsor;
- Data em que o primeiro resultado de pouso bem-sucedido na plataforma de solo foi alcançado;
- Nomes dos propulsores que tiveram sucesso na nave drone e têm massa de carga útil entre 4000 e 6000 kg;
- Número total de resultados de missão bem-sucedidos e com falha;
- Nomes das versões de propulsor que transportaram a massa máxima de carga útil;
- Resultados de pouso com falha na nave drone, suas versões de propulsor e nomes de locais de lançamento para o ano de 2015; e
- Classificação da contagem de resultados de pouso (como Falha (nave drone) ou Sucesso (plataforma de solo)) entre a data 2010-06-04 e 2017-03-20

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

```
%sql SELECT * from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;  
* sqlite:///my_data1.db
```

Build an Interactive Map with Folium

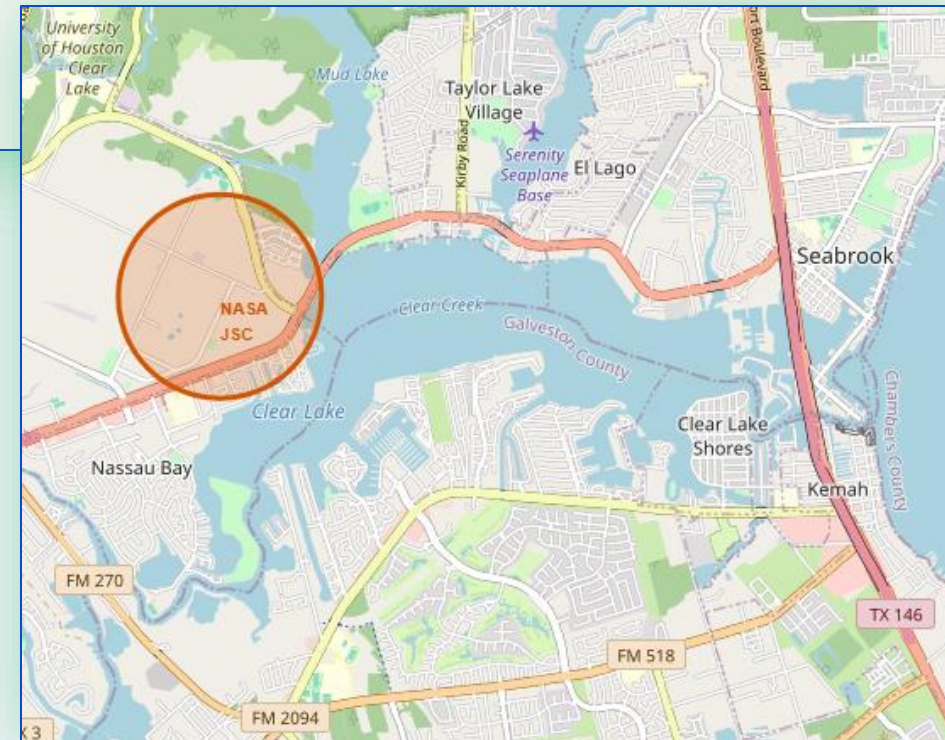
A taxa de sucesso do lançamento pode depender de muitos fatores, como massa da carga útil, tipo de órbita e assim por diante. Também podemos depender da localização e proximidade de um local de lançamento, ou seja, a posição inicial das trajetórias do foguete. Encontrar um local ideal para construir um local de lançamento certamente envolve muitos fatores e, com sorte, poderíamos descobrir alguns dos fatores analisando os locais existentes do local de lançamento.

Com a análise exploratória de dados podemos visualizar o conjunto de dados de lançamento da SpaceX usando matplotlib e seaborn e descobrimos algumas correlações preliminares entre o local de lançamento e as taxas de sucesso.

Neste laboratório, realizamos a análises visuais mais interativas usando o Folium.

<https://github.com/gdmacedo/applied-data-science-capstone/blob/main/3.2%20-%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb>

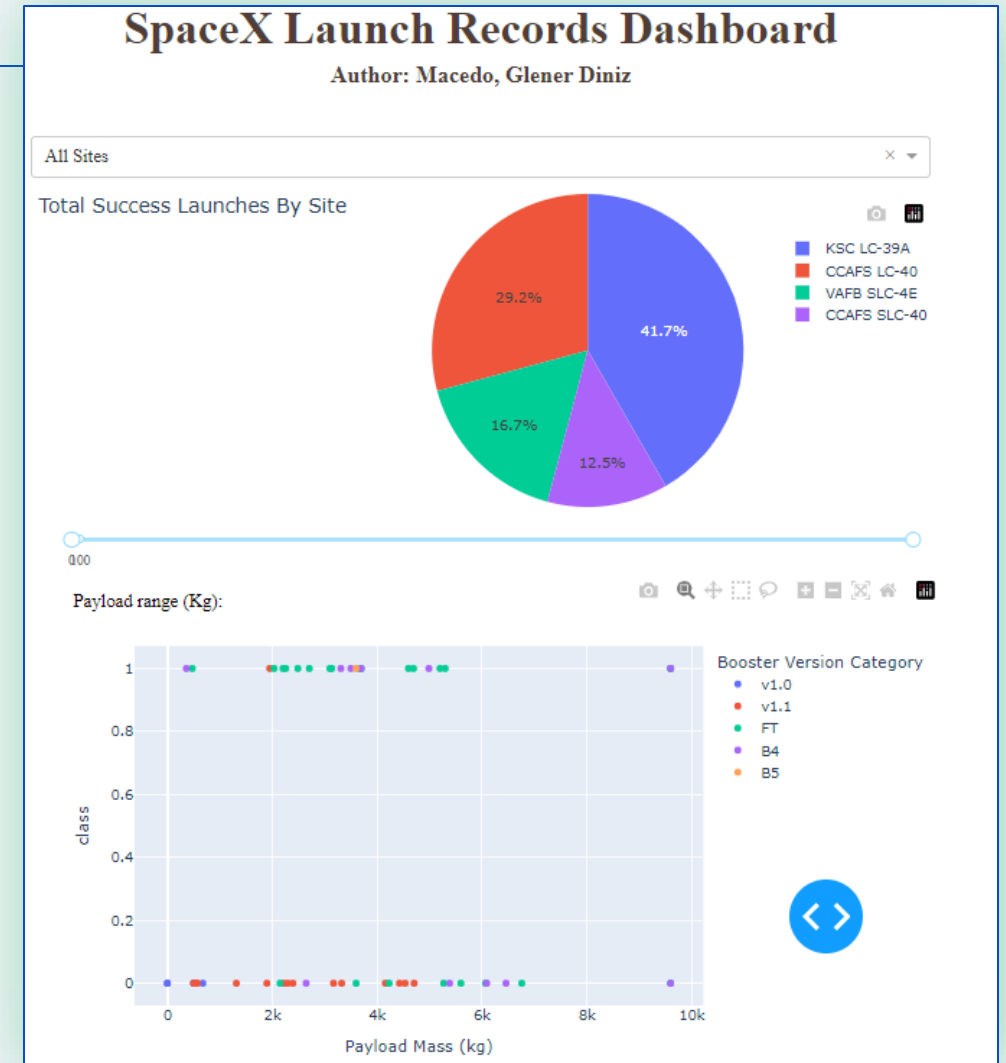
```
# Create a blue circle at NASA Johnson Space Center's coordinate with a popup Label showing its name
circle = folium.Circle(nasa_coordinate, radius=1000, color='#d35400',
                      fill=True).add_child(folium.Popup('NASA Johnson Space Center'))
# Create a blue circle at NASA Johnson Space Center's coordinate with a icon showing its name
marker = folium.map.Marker(nasa_coordinate,
                          # Create an icon as a text Label
                          icon=DivIcon(icon_size=(20,20), icon_anchor=(0,0),
                                      html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % 'NASA JSC',
                                      ),
                          )
site_map.add_child(circle)
site_map.add_child(marker)
```



Build a Dashboard with Plotly Dash

Neste laboratório, criou-se um aplicativo Plotly Dash para realizar análises visuais interativas sobre os dados de lançamento da SpaceX em tempo real. Contendo componentes de entrada, como uma lista suspensa e um controle deslizante de intervalo para interagir com um gráfico de pizza e um gráfico de ponto de dispersão. Construímos este aplicativo de painel por meio das seguintes etapas:

1. Componente de entrada suspenso do site de lançamento
2. Função de retorno de chamada para renderizar o gráfico de pizza de sucesso com base no menu suspenso do site selecionado
3. Controle deslizante de intervalo para selecionar a carga útil
4. Função de retorno de chamada para renderizar o gráfico de dispersão success-payload-scatter-chart



<https://github.com/gdmacedo/applied-data-science-capstone/tree/main/3.3%20-%20Build%20an%20Interactive%20Dashboard%20with%20Plotly%20Dash>

Predictive Analysis (Classification)

Com um total de quatro modelos de classificação foram comparados:

- regressão logística;
- máquina de vetores de suporte;
- árvore de decisão e
- k nearest neighbors.

Examinando a matriz de confusão, vemos que a regressão logística pode distinguir entre as diferentes classes.

Vemos que o problema são os falsos positivos.

Visão geral:

- Verdadeiro Positivo – 12 (O rótulo Verdadeiro é aterrado, o rótulo Previsto também é aterrado)
- Falso Postivo – 3 (o rótulo verdadeiro não foi obtido, o rótulo previsto foi obtido)

```
def plot_confusion_matrix(y,y_predict):  
    "Esta função plota a matriz de confusão"  
    from sklearn.metrics import confusion_matrix  
  
    cm = confusion_matrix(y, y_predict)  
    ax= plt.subplot()  
    sns.heatmap(cm, annot=True, ax = ax); #annot=True para anotar células  
    ax.set_xlabel('Rótulos previstos')  
    ax.set_ylabel('Rótulos verdadeiros')  
    ax.set_title('Matriz de Confusão');  
    ax.xaxis.set_ticklabels(['Não Aprovado', 'Aprovado']);  
    ax.yaxis.set_ticklabels(['Não Aprovado', 'Aprovado'])  
    plt.show()
```

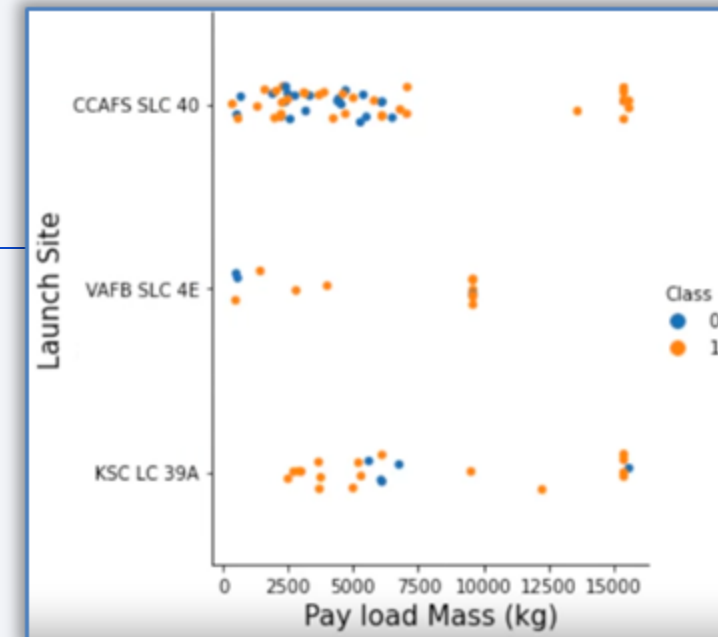


<https://github.com/gdmacedo/applied-data-science-capstone/blob/main/4.1.2%20-%20Complete%20the%20Machine%20Learning%20Prediction%20lab.ipynb>

Results

Análise Exploratória de Dados

- Foram usados quatro locais de lançamento diferentes pela A Space X;
- Os primeiros lançamentos foram feitos para a própria Space X e para a NASA;
- A carga útil média do propulsor F9 v1.1 é de 2.928 kg;
- O primeiro resultado de pouso bem-sucedido aconteceu em 2015, cinco anos após o primeiro lançamento;
- Muitas versões do propulsor Falcon 9 foram bem-sucedidas no pouso em naves drone com carga útil acima da média;
- Quase 100% dos resultados da missão foram bem-sucedidos;
- Duas versões do propulsor falharam no pouso em naves drone em 2015: F9 v1.1 B1012 e F9 v1.1 B1015;
- O número de resultados de pouso melhorou com o passar dos anos.



Results

Demonstração analítica interativa em capturas de tela

- Com o uso da análise interativa foi possível identificar os locais de lançamento onde costumam estar em locais seguros, por exemplo perto do mar, de modo que tenhaos uma boa infraestrutura logística ao redor, para se ter acesso fácil.
- Os lançamentos na sua maioria aconteceram em locais de mais próximos possível a linha do equador, na costa leste.

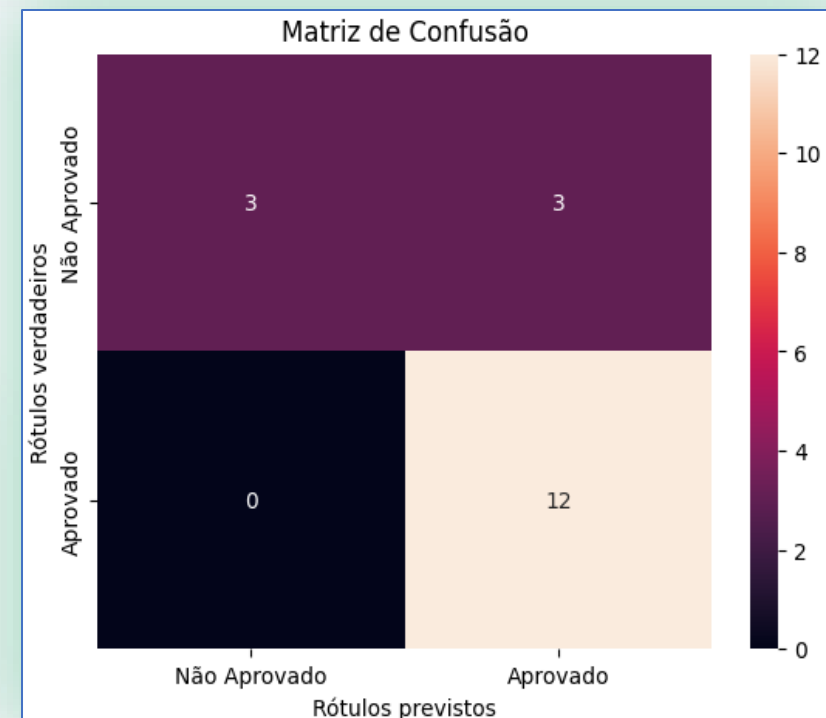
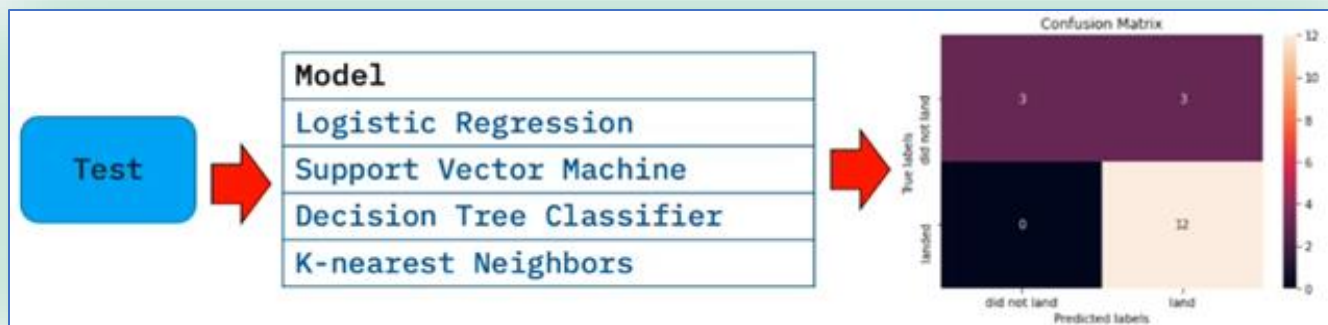


Results

Resultados da análise preditiva

A análise preditiva mostrou que a Decision Tree Classifier é o melhor modelo para prever pousos bem-sucedidos, tendo precisão acima de 87% e precisão para dados de teste acima de 94%. Obtemos com precisão de knn_cv nos dados de teste usando o método score, descobrindo assim o método que tem melhor desempenho.

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





Section 2

Insights drawn from EDA

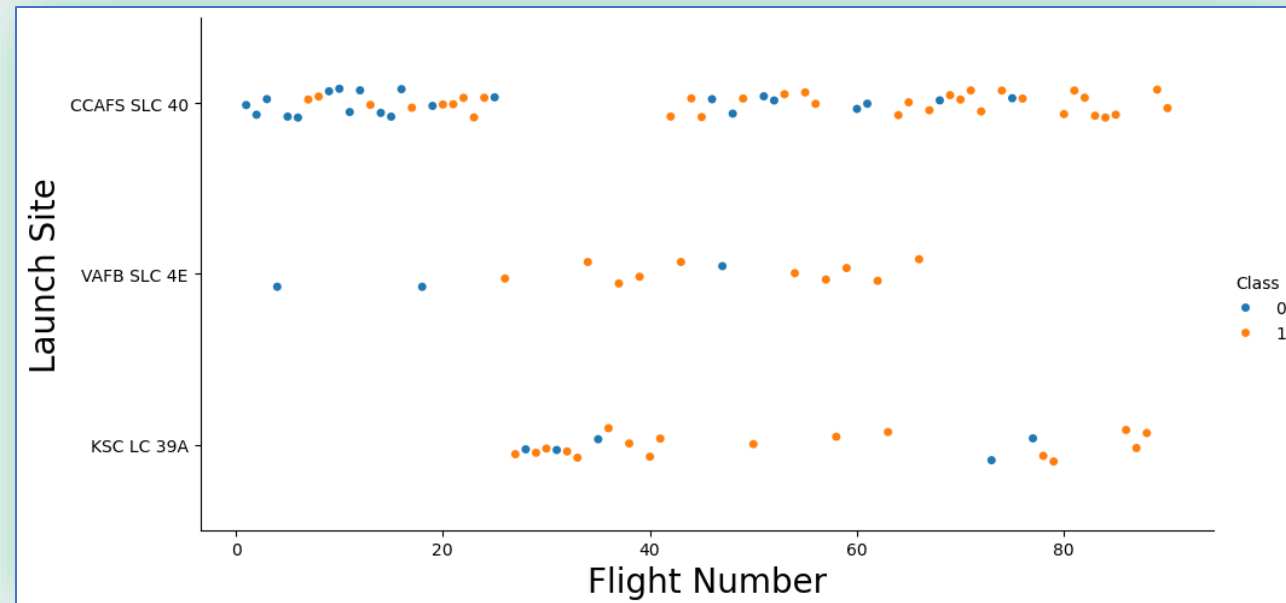
Flight Number vs. Launch Site

Usando a função catplot para plotar FlightNumber vs LaunchSite, podemos definir o parâmetro x para FlightNumber, defina o y para Launch Site e definimos o parâmetro hue para 'class'.

Gráfico de pontos de dispersão com o eixo x para ser o número do voo e o eixo y para ser o local de lançamento e o matiz para ser o valor da classe.

Conforme podemos ver os pontos em azul que representa as falhas, observamos que os voos anteriores tiveram uma taxa de sucesso menor. E os voos posteriores tiveram uma taxa de sucesso maior representados pelos pontos em laranja.

```
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect=2)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
plt.show()
```



Consideramos também que cerca da metade dos lançamentos foram do local de lançamento CCAFS SLC 40, sendo que AFB SLC 4E e KSC LC 39A tiveram taxas de sucesso maiores.

Portanto concluímos que novos lançamentos têm uma taxa de sucesso maior.

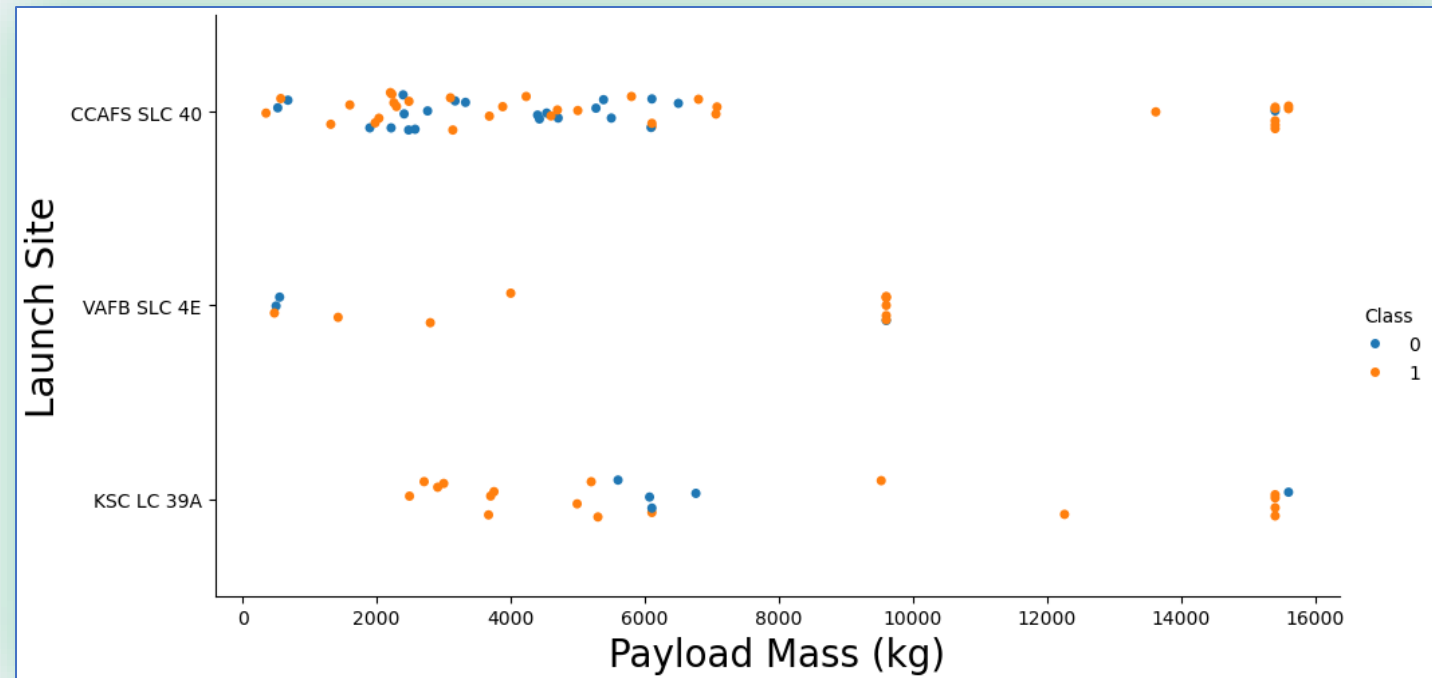
Payload vs. Launch Site

Visualizamos a relação entre a massa da carga útil e o local de lançamento. Também podemos observar se há alguma relação entre os locais de lançamento e sua massa de carga útil.

Nesse gráfico de pontos de dispersão com o eixo x para ser a massa de carga útil (kg) e o eixo y para ser o local de lançamento e a matriz para ser o valor da classe. Normalmente, quanto maior a massa da carga útil (kg), maior a taxa de sucesso.

A maioria dos lançamentos com uma carga útil maior que 7.000 kg foram bem-sucedidos KSC LC 39A tem uma taxa de sucesso de 100% para lançamentos menores que 5.500 kg VAFB SKC 4E não lançou nada maior que ~10.000 kg.

```
sns.catplot(y = "LaunchSite", x = "PayloadMass", hue = "Class", data = df, aspect=2)  
plt.xlabel("Payload Mass (kg)", fontsize = 20)  
plt.ylabel("Launch Site", fontsize = 20)  
plt.show()
```



Success Rate vs. Orbit Type

Podemos visualizar a relação entre a taxa de sucesso de cada tipo de órbita, de modo que em seguida, verificamos visualmente se há alguma relação entre a taxa de sucesso e o tipo de órbita.

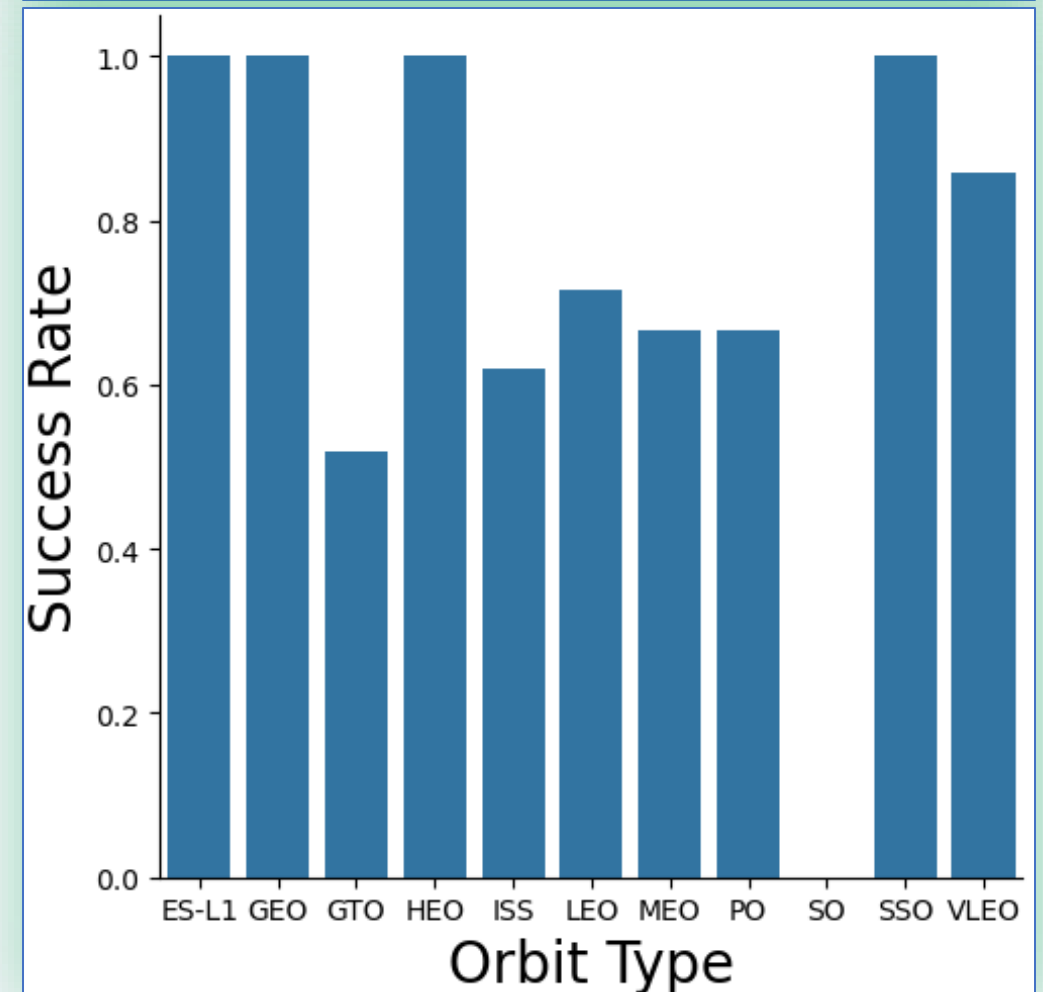
Usando o método groupby na coluna Orbit e obtenha a média da coluna Class, onde obtemos os seguintes resultados:

- 100% Taxa de sucesso: ES-L1, GEO, HEO e SSO
- 50%-80% Taxa de sucesso: GTO, ISS, LEO, MEO, PO
- 0% Taxa de sucesso: SO

Resultando nas maiores taxas de sucesso acontecem nas órbitas ES-L1, GEO, HEO e SSO.

Seguido por VLEO (acima de 80%); e LFO (acima de 70%).

```
sns.catplot(x = 'Orbit', y = 'Class',  
            data = df.groupby('Orbit')['Class'].mean().reset_index(),  
            kind = 'bar')  
plt.xlabel('Orbit Type',fontsize=20)  
plt.ylabel('Success Rate',fontsize=20)  
plt.show()
```



Flight Number vs. Orbit Type

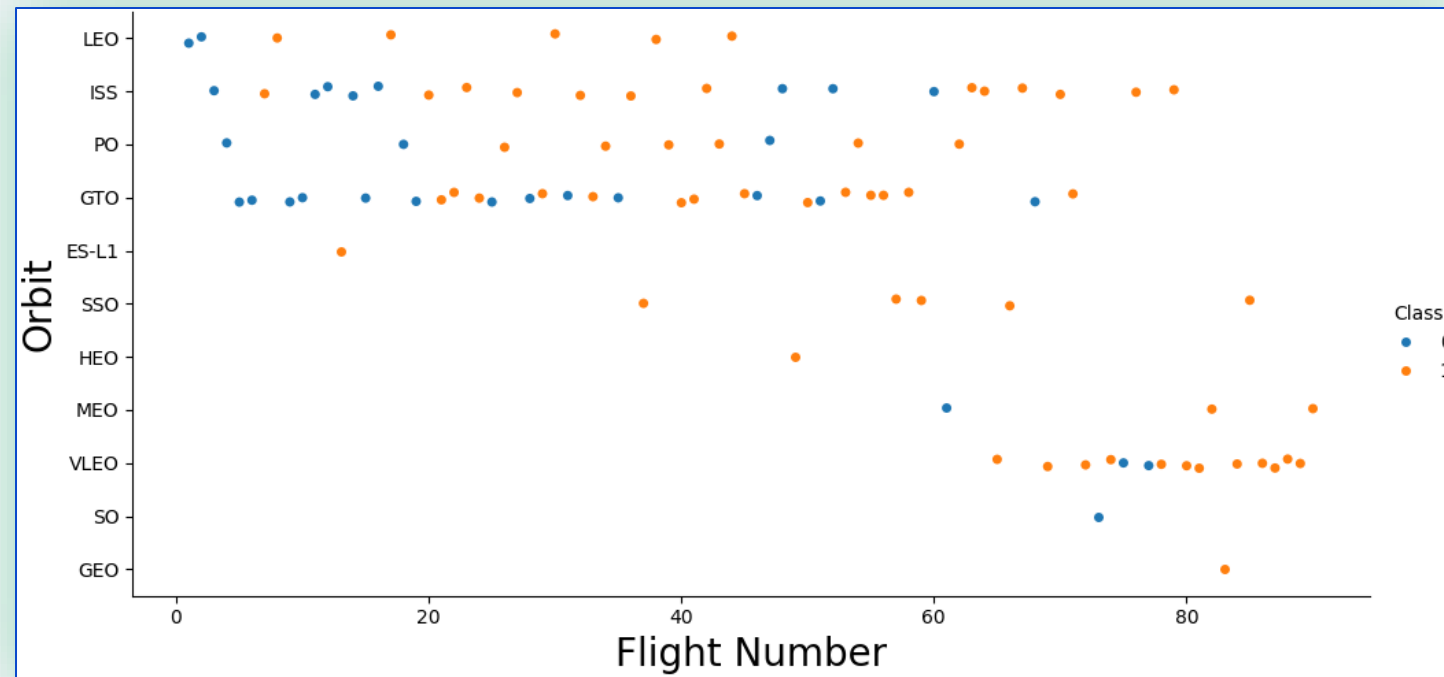
Aparentemente, a taxa de sucesso melhorou ao longo do tempo para todas as órbitas;

- A órbita VLEO parece uma nova oportunidade de negócio, devido ao recente aumento de sua frequência.

```
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect=2)
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```

Para cada órbita, podemos ver se há alguma relação entre FlightNumber e tipo de órbita. O gráfico de pontos de dispersão com o eixo x para ser o FlightNumber e o eixo y para ser a Órbita, e o matiz para ser o valor da classe.

Observe que na órbita LEO, o sucesso parece estar relacionado ao número de voos. Por outro lado, na órbita GTO, parece não haver relação entre número de voos e sucesso.



Payload vs. Orbit Type

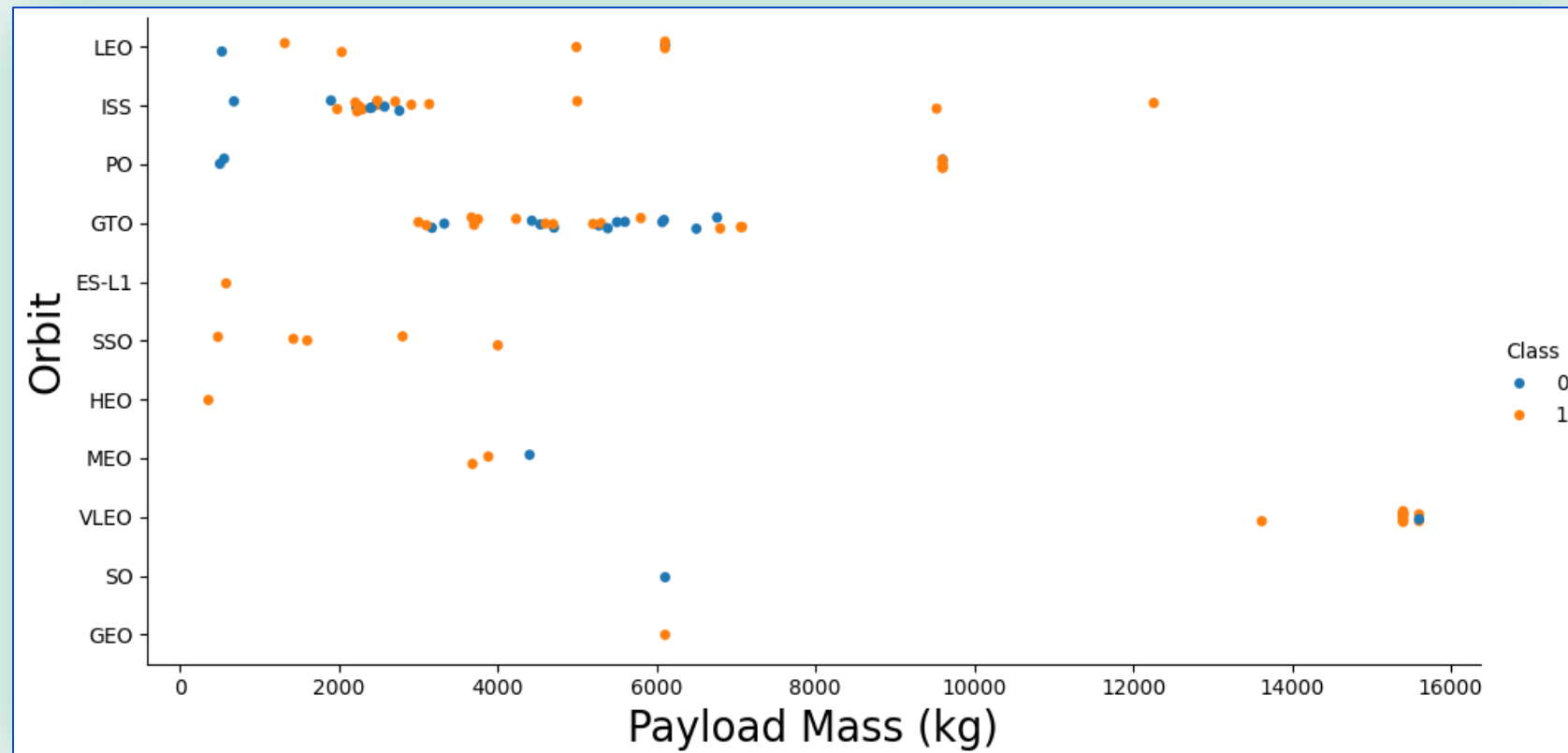
Analizando os Pontos de dispersão de carga útil vs. tipo de órbita, podemos visualizar a relação entre a massa da carga útil e o tipo de órbita.

Da mesma forma, podemos traçar os gráficos de pontos de dispersão da massa da carga útil versus a órbita para revelar a relação entre a massa da carga útil e o tipo de órbita.

Assim, apresentamos um gráfico de pontos de dispersão com o eixo x para ser a massa da carga útil e o eixo y para ser a órbita e o matiz para ser o valor da classe.

Cargas úteis pesadas são melhores com órbitas LEO, ISS e PO e a órbita GTO tem sucesso misto com cargas úteis mais pesadas.

```
sns.catplot(y = "Orbit", x = "PayloadMass",  
            hue = "Class", data = df, aspect= 2)  
plt.xlabel("Payload Mass (kg)", fontsize=20)  
plt.ylabel("Orbit", fontsize=20)  
plt.show()
```



Launch Success Yearly Trend

Traçamos um gráfico de linhas com o eixo x representando o ano e o eixo y representando a taxa média de sucesso para obter a tendência média de sucesso do lançamento.

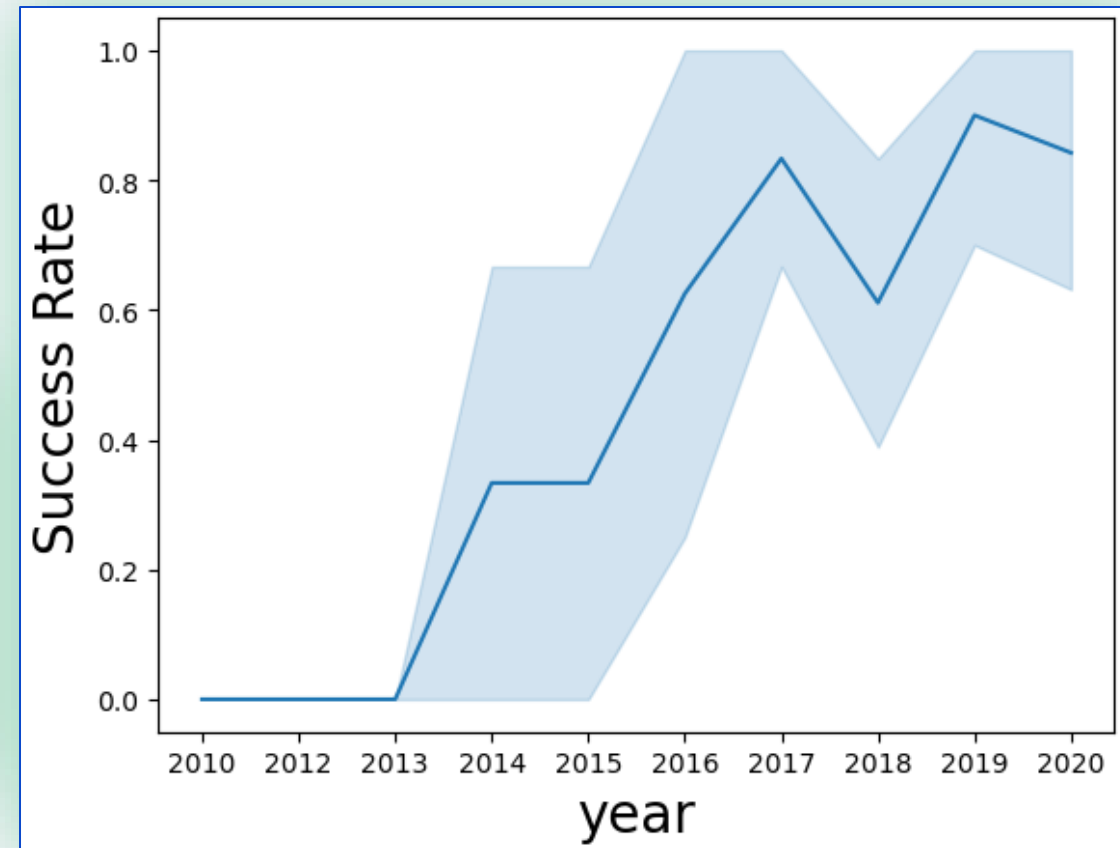
Com a primeira função `Extract_year()`, que ajuda a obter o ano a partir da data e um gráfico de linhas com o eixo x para ser o ano extraído e o eixo y para ser a taxa de sucesso.

Com podemos concluir que a taxa de sucesso :

- Aumentou de 2013-2017 e 2018-2019;
- Diminuiu de 2017-2018 e de 2019-2020 e
- No geral, melhorou desde 2013.

```
year=[]
def Extract_year():
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
Extract_year()
df['Date'] = year
df.head()
```

```
sns.lineplot(data = df,
             x = "Date",
             y = "Class")
plt.xlabel("year",
           fontsize = 20)
plt.ylabel("Success Rate",
           fontsize=20)
plt.show()
```



All Launch Site Names

Obtendo alguns insights preliminares sobre como cada variável importante afeta

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

Selecionaremos os recursos que serão usados na previsão de sucesso no módulo futuro.

Usamos a função `get_dummies` e `features dataframe` para aplicar `OneHotEncoder` à coluna `Orbits`, `LaunchSite`, `LandingPad` e `Serial`.

Atribindo assim o valor à variável `features_one_hot`, exiba os resultados usando o método `head`. Neste dataframe de resultados deve incluir todos os recursos, incluindo os codificados.

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

LaunchSite
CCAFS SLC 40
CCAFS SLC 40
CCAFS SLC 40
VAFB SLC 4E
CCAFS SLC 40

Launch Site Names Begin with 'CCA'

Conforme instrução SQL, apresentamos os cinco registros onde os locais de lançamento começam com 'CCA'.

```
%sql SELECT * from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;

* sqlite:///my_data1.db
```

Tendo como resultado da sua consulta SQL os registros abaixo:

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Total Payload Mass

Para calcular a massa total da carga útil transportada pelos propulsores lançados pela NASA (CRS) :

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
```

Onde obtemos o valor de **45.596 kg** (total) transportados por propulsores lançados pela NASA (CRS).

payloadmass
45596

E tendo como massa média de carga transportada pelo propulsor versão F9 v1.1 o valor de 2.928 kg (média), obtidos conforme a instrução SQL:

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

payloadmass
2928.4

Average Payload Mass by F9 v1.1

Massa média de carga transportada pelo propulsor versão F9 v1.1:

Selecionando/Filtrando os dados pela versão de reforço acima e calculando a massa média da carga útil, obtivemos o valor de 2.928 kg.

```
%sql SELECT avg(PAYLOAD_MASS_KG_) as 'Avg Payload (kg)' FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

Apresente o resultado da sua consulta com uma breve explicação:

Avg Payload (kg)

2928.4

First Successful Ground Landing Date

Encontramos a data do primeiro resultado de pouso bem-sucedido na plataforma de solo, de acordo com a seguinte instrução SQL.

```
%sql select min(DATE) as 'Min Date' from SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

Ao filtrar os dados pelo resultado do pouso bem-sucedido na plataforma de solo e obter o valor mínimo para a data, é possível identificar a primeira ocorrência, que aconteceu em 22 de dezembro de 2015.

Min Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Os propulsores que pousaram com sucesso no navio drone e tinham massa de carga útil maior que 4000, mas menor que 6000 pode ser selecionado conforme as instruções SQL:

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME='Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

Selecionando versões distintas de booster de acordo com os filtros acima, estes 4 são o resultado.

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

Total Number of Successful and Failure Mission Outcomes

Calculando o número total de resultados de missões bem-sucedidas e falhas:

```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number from SPACEXTBL GROUP BY MISSION_OUTCOME order by MISSION_OUTCOME;
```

Agrupando os resultados da missão e contando os registros de cada grupo nos levou ao resumo abaixo.

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

Apresentamos os nomes das booster_versions que carregaram a massa máxima de carga útil, através de uma subconsulta.

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL  
      where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

Estes são os propulsores que transportaram a massa máxima de carga registrada no conjunto de dados.

boosterversion
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

Registros que exibirão os nomes dos meses, resultados de pousos com falha, mal sucedidos no navio drone, versões do booster, launch_site para os meses do ano de 2015.

```
%sql SELECT substr(Date, 6,2) as Month, Landing_Outcome as 'Landing Outcome', Booster_Version as 'Booster Version', Launch_Site as 'Launch Site'  
FROM SPACEXTABLE WHERE substr(Date,0,5)='2015' and Landing_Outcome = 'Failure (drone ship)'
```

Com apenas duas ocorrências temos:

Month	Landing Outcome	Booster Version	Launch Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Classificando a contagem de resultados de pouso com falha em navio drone ou com Sucesso em plataforma de pouso, entre as datas 4 de junho de 2010 a 20 de março de 2017, em ordem decrescente.

```
%sql SELECT LANDING_OUTCOME, count(*) as count_outcomes FROM SPACEXTBL WHERE DATE  
BETWEEN '2010-06-04' AND '2017-03-20' group by [Landing_Outcome]  
ORDER by count_outcomes DESC;
```

Landing_Outcome	count_outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

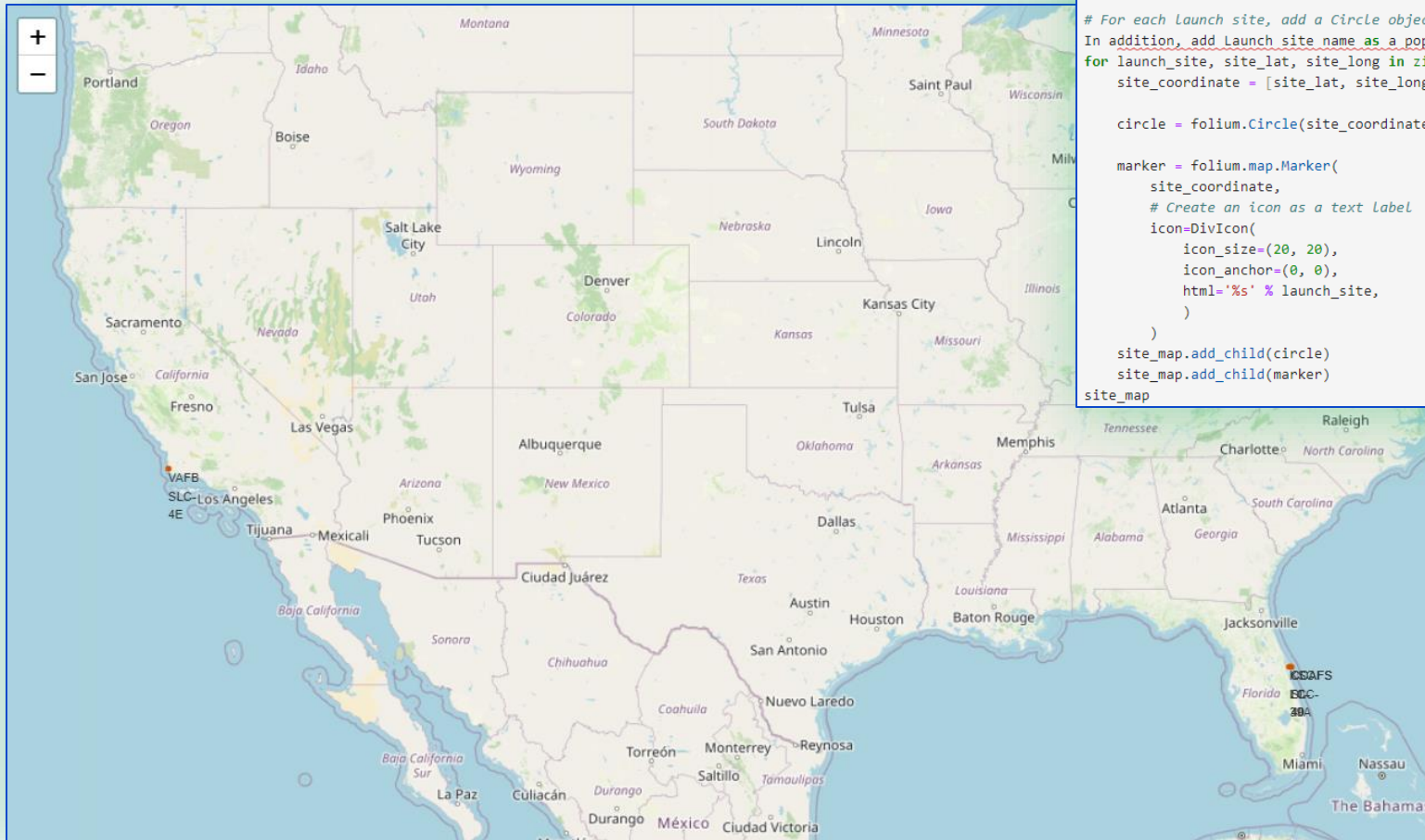
A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

Launch Sites Proximities Analysis

All launch sites

Locais de lançamento são próximos ao mar, provavelmente por segurança, mas não muito longe de estradas e ferrovias.



```
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)

# For each Launch site, add a Circle object based on its coordinate (Lat, Long) values.
In addition, add launch site name as a popup label
for launch_site, site_lat, site_long in zip(launch_sites_df['Launch Site'], launch_sites_df['Lat'], launch_sites_df['Long']):
    site_coordinate = [site_lat, site_long]

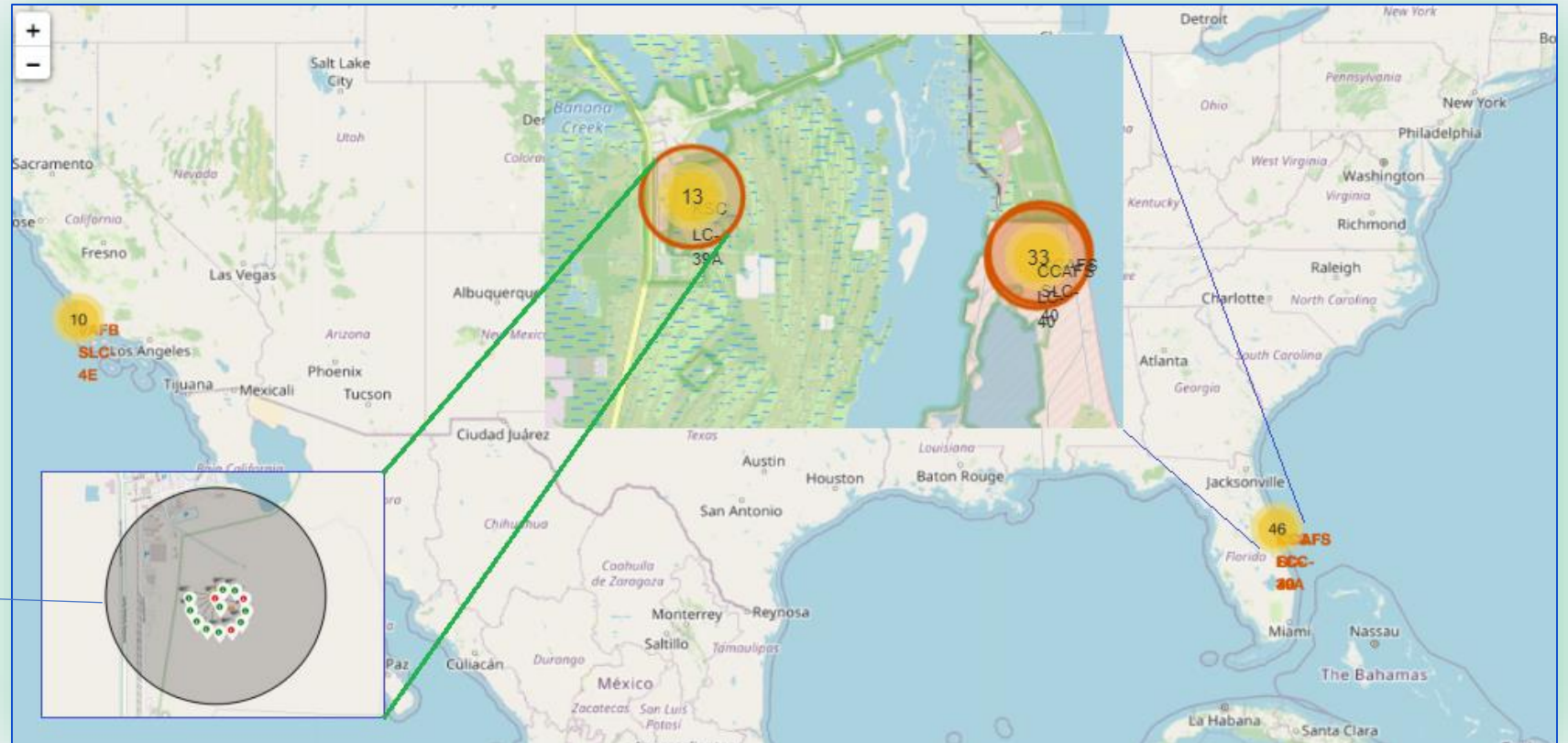
    circle = folium.Circle(site_coordinate, radius=1000, color='#d35400', fill=True).add_child(folium.Popup(launch_site))

    marker = folium.map.Marker(
        site_coordinate,
        # Create an icon as a text label
        icon=DivIcon(
            icon_size=(20, 20),
            icon_anchor=(0, 0),
            html='%s' % launch_site,
        )
    )
    site_map.add_child(circle)
    site_map.add_child(marker)
site_map
```

Launch Outcomes by Site

Exemplo de resultados do lançamento do local de lançamento do KSC LC-39A

Os marcadores verdes indicam sucesso e os vermelhos indicam falha.

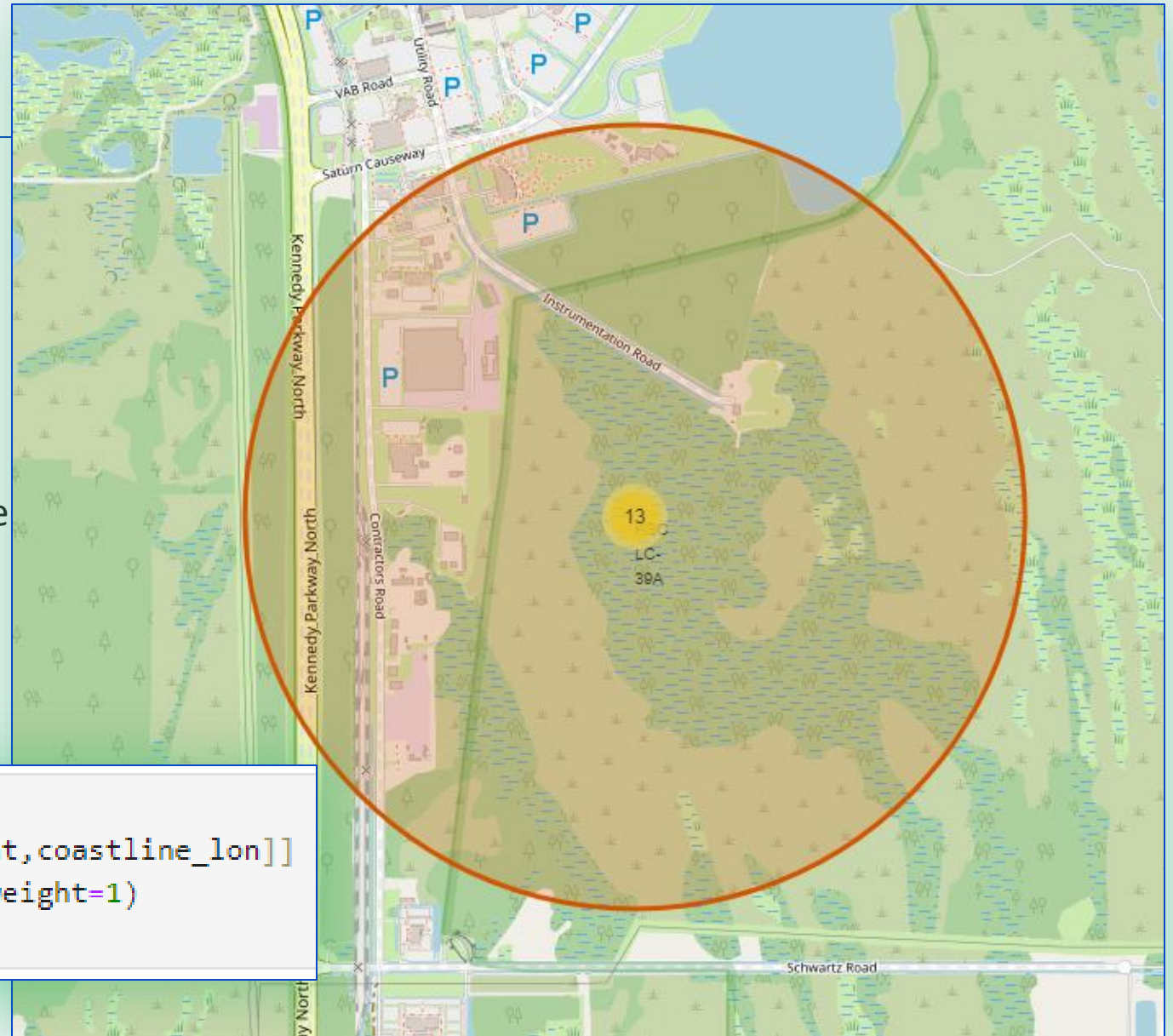


Logistics and Safety

Destaque para o mapa Folium gerado de modo que podemos ver a captura de tela de um local de lançamento selecionado para suas proximidades, como ferrovia, rodovia, litoral, com a distância calculada e exibida.

O local de lançamento do KSC LC-39A tem bons aspectos logísticos, estando perto de ferrovias e estradas e relativamente longe de áreas habitadas. Com um objeto `folium.PolyLine` usando as coordenadas do litoral e as coordenadas do local de lançamento.

```
coordinates = [[launch_site_lat,  
                launch_site_lon],[coastline_lat,coastline_lon]]  
lines=folium.PolyLine(locations=coordinates, weight=1)  
site_map.add_child(lines)
```





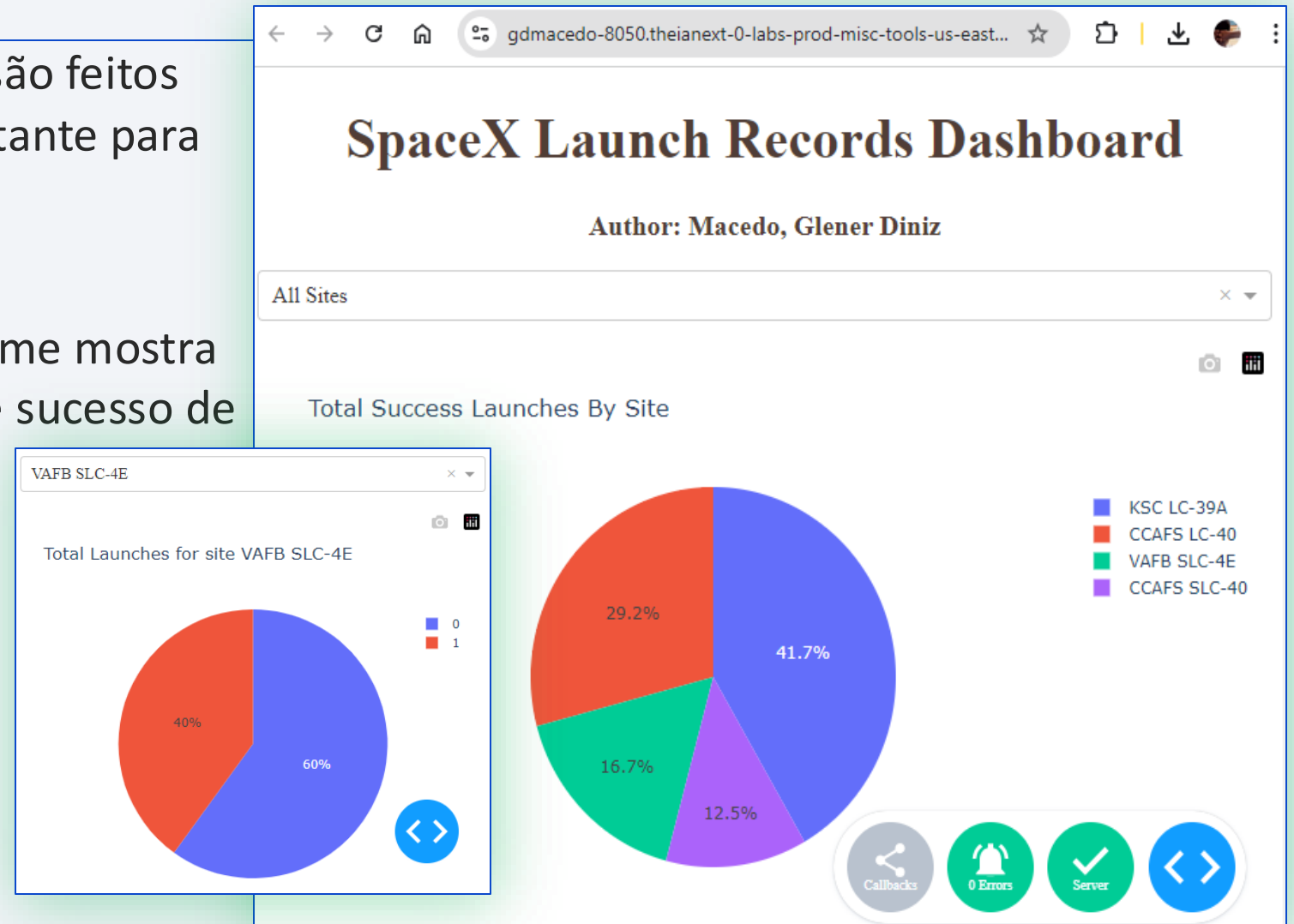
Section 4

Build a Dashboard with Plotly Dash

Successful Launches by Site

O local de onde os lançamentos são feitos parece ser um fator muito importante para o sucesso das missões.

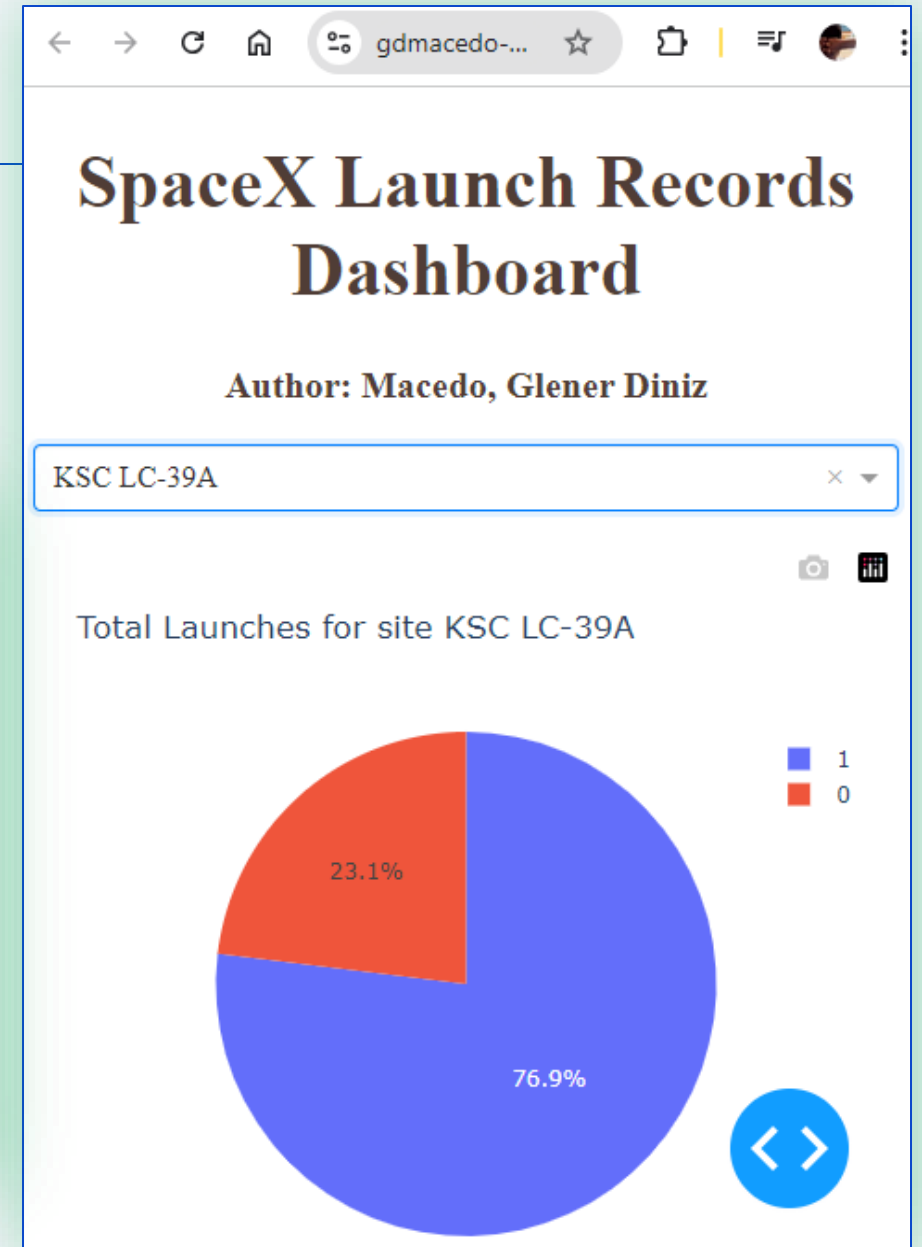
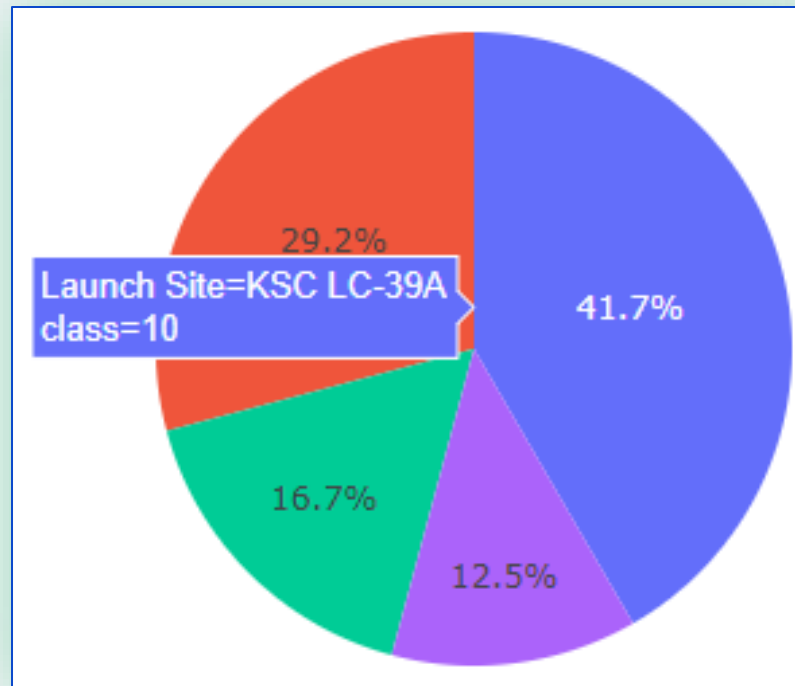
Podendo selecionar todos ou um determinado lançamento, conforme mostra a captura de tela da contagem de sucesso de lançamento para todos os locais, apresentado nos gráfico de pizza.



Launch Success - KSC LC-29A

De acordo com os quatro locais, o local de maior sucesso foi o KSC LC-29A, onde o sucesso como porcentagem do total temos a maior taxa de sucesso entre os locais de lançamento (76,9%), ou seja dez lançamentos bem-sucedidos e três lançamentos mal sucedidos.

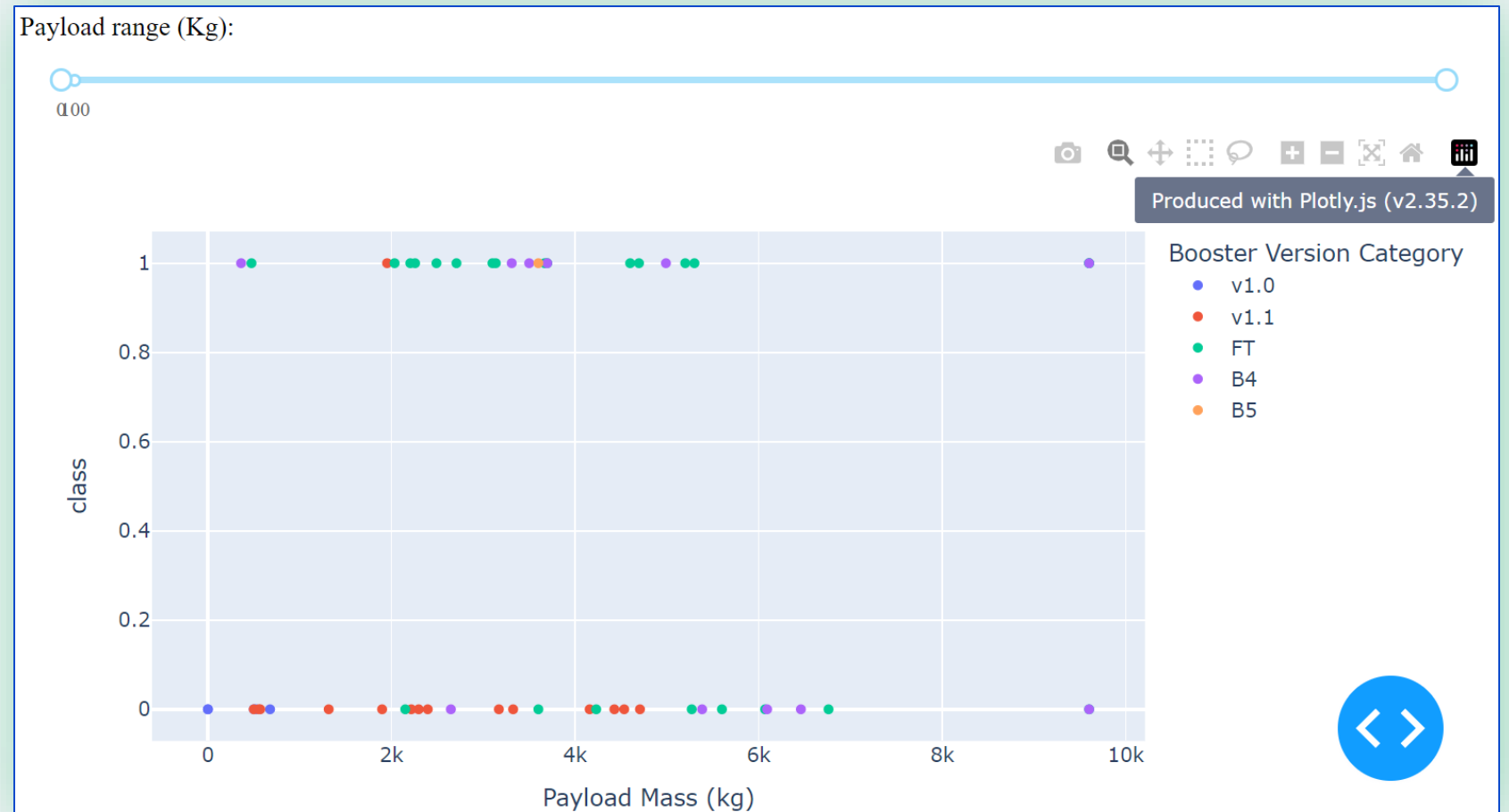
Notem que o KSC LC-29A, representa, 41,7% em relação ao total de lançamentos.



Payload vs. Launch Outcome

Gráfico de dispersão de Carga útil vs. Resultado do lançamento para todos os locais, com diferentes cargas úteis selecionadas no controle deslizante de intervalo

Cargas úteis abaixo de 6.000 kg e reforços FT são a combinação mais bem-sucedida.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

Entre os quatros modelos de classificação foram testados, e suas precisões são plotadas ao lado.

Decision Tree Classifier, é o modelo com a maior precisão de classificação, que tiveram as precisões acima de 87%.

Todos os modelos tiveram desempenho aproximadamente no mesmo nível e tiveram as mesmas pontuações e precisão. Isso se deve provavelmente ao pequeno conjunto de dados.

```
models = {'KNeighbors': knn_cv.best_score_,
          'DecisionTree': tree_cv.best_score_,
          'LogisticRegression': logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

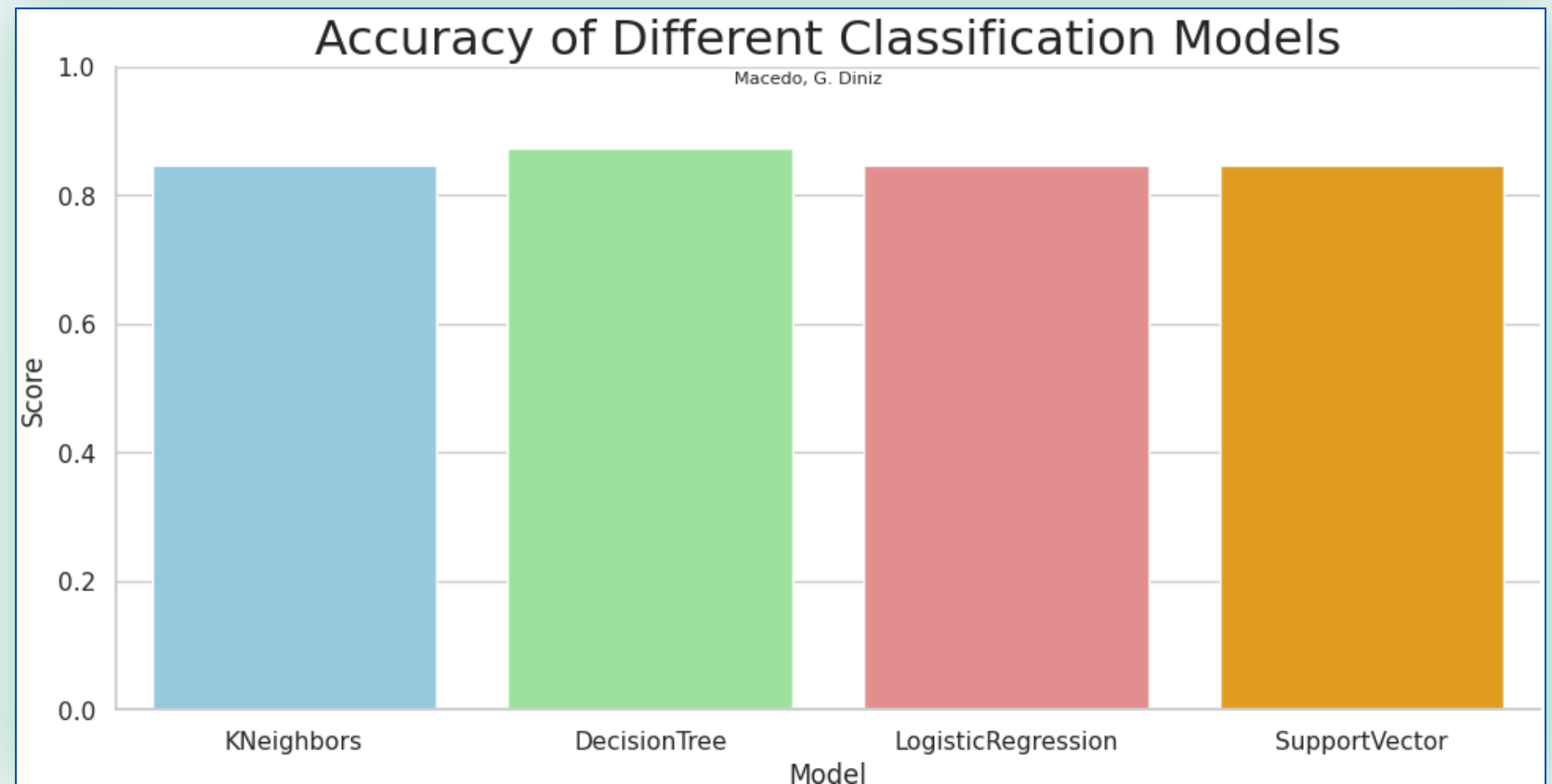
for model in models:
    print(model, ":", models[model])

bestalgorithm = max(models, key=models.get)

print('\nBest model is',
      bestalgorithm, 'with a score of',
      models[bestalgorithm])

KNeighbors : 0.8482142857142858
DecisionTree : 0.875
LogisticRegression : 0.8464285714285713
SupportVector : 0.8482142857142856

Best model is DecisionTree with a score of 0.875
```



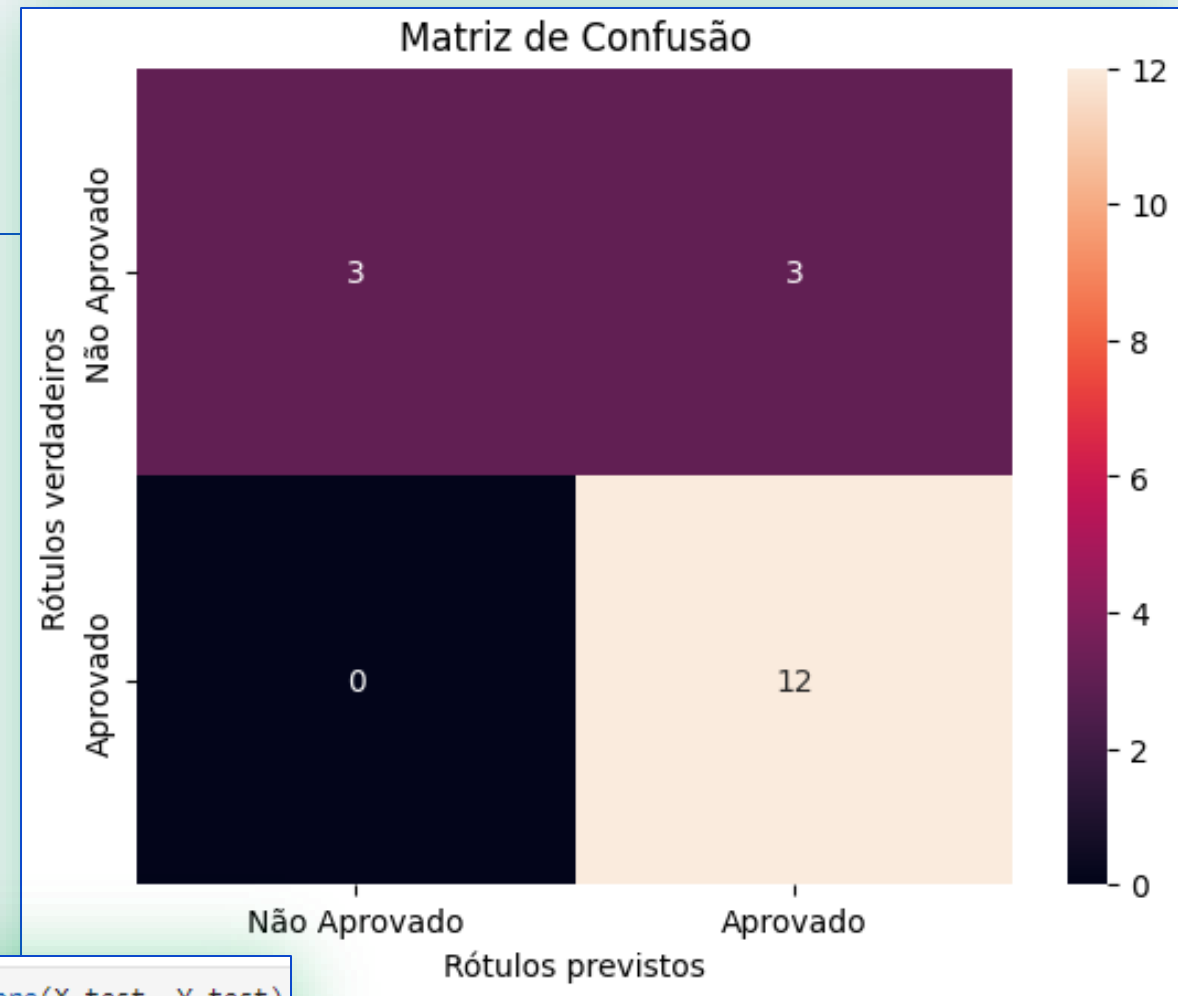
Confusion Matrix

A matriz de confusão do Decision Tree Classifier prova sua precisão ao mostrar os grandes números de verdadeiros positivos e verdadeiros negativos em comparação com os falsos.

Temos nessa matriz:

- 12 verdadeiros positivos;
- 3 verdadeiros negativos;
- 3 falsos positivos e ;
- 0 falsos negativos.

Calculamos a precisão de knn_cv nos dados de teste usando o método score e e traçamos este gráfico.



```
knn_cv_score = knn_cv.score(X_test, Y_test)  
print("score :", knn_cv_score)
```

```
score : 0.8333333333333334
```

Podemos traçar a matriz de confusão

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

Conclusions

Os modelos tiveram desempenho semelhante no conjunto de teste com o modelo **Decision Tree** superando ligeiramente, onde a teve a maior pontuação de precisão de dados de treinamento, enquanto todos os modelos tiveram a mesma pontuação de precisão de dados de teste. (incluindo alguns falsos positivos).

A maioria dos **locais de lançamento** fica perto da costa, juntamente da linha do equador para um impulso natural adicional, devido à velocidade de rotação da Terra e o que arcar com o custo menor de colocar combustível extra e propulsores.

A versão Booster com mais lançamentos e a maior taxa de sucesso é a **FT** e ao longo do tempo temos mais experiência, e obtemos assim mais sucesso nos lançamentos.

A maior **taxa de sucesso** entre os locais de lançamento, foi obtido com o **KSC LC-39-A**, tendo uma taxa de sucesso de 100% para lançamentos com menos de 5.500 kg.

Teve sucesso de 100% as **Órbitas** ES-L1, GEO, HEO e SSO.

Appendix

IBM – Python Capstone Project Repo

Sete notebooks Jupyter, código de painel e um deck

- 1 - [Complete the Data Collection API Lab](#)
- 2 - [Data Collection with Web Scraping](#)
- 3 – [Labs jupyter SpaceX Data Wrangling](#)
- 4 - [Jupyter Labs EDA SQL sqllite](#)
- 5 - [EDA with Data Visualization](#)
- 6 - [EDA com laboratório de visualização - edadataviz](#)
- 7 – [Jupyter Labs EDA SQL Coursera SQLite](#)
- 8 - [Interactive Visual Analytics with Folium](#)
- 9 - [Build an Interactive Dashboard with Plotly Dash](#)
- 10 - [DV0101EN-3-5-1-Generating-Maps-in-Python-py-v2.0](#)
- 11 - [Complete the Machine Learning Prediction lab](#)

Thank you!

