Rocketing Through Data

Rocketing Through Data: An In-Depth Analysis of Space Mission Statistics

Introduction

Space exploration has seen significant growth from 1995 to 2020, driven by both government and private entities. This project analyzes global space mission data from this period, focusing on key factors such as mission frequency, country involvement, costs, and success rates.

With the potential asteroid threat predicted for 2038, this report also aims to assess which organisations are best equipped for asteroid impact prevention and prepare for planetary defence. By uncovering patterns and trends in space exploration, the project will provide insights into future missions and preparedness for celestial threats.

Keywords: mission frequency, country involvement, costs, success rates, asteroid impact, planetary defence, patterns, trends and celestial threats.

Objectives

Analyze Temporal Trends : Identify patterns and variations in global space launches over time, highlighting trends and seasonal shifts from 1995 to 2020.

Evaluate Company Performance: Assess the performance of space companies based on launch frequency, mission types, and success rates.

Geographical Analysis: Explore the distribution of launch sites worldwide and examine how location influences mission outcomes.

Dataset Overview:

The dataset titled ‘SpaceProject.csv’ consists of 8 columns, which are as follows:

- 'Unnamed: 0.1'
- 'Unnamed: 0'
- 'Company Name'
- 'Location'
- 'Datum'
- 'Detail'
- 'Status Rocket'
- 'Rocket'
- 'Status Mission'

Data Preparation:

Step 1: Loading the Dataset

The dataset, named "SpaceProject.xlsx," was loaded into the environment using the Pandas library.

(insert code)

Step 2: Column Inspection

The columns were inspected to understand the data structure. The key variables for predictive modeling were noted, such as mission success rates, costs, and the specific countries involved in space missions.

(insert code)

Step 3: Handling Missing Data

After an initial overview of the dataset, the presence of missing values was checked using the following code:

(insert code)

This revealed the number and percentage of missing values in each column. Understanding this was critical to ensure that no key information was missing that might affect the predictive model.

Step 4: Data Overview

The function provided an overview of the dataset, including data types and non-null counts, ensuring the data was ready for analysis.

(insert code)

After preparing the data and understanding the overview of the data,we further clean the data to ensure accuracy and consistency in the data required to perform the necessary analysis.

Data Cleaning:

Step 1:Dropping Irrelevant Columns

The columns **‘Unnamed: 0.1’** and **‘Unnamed: 0’** were found to be **irrelevant to the analysis**. These columns likely represent index or serial numbers that have no contextual significance to our study. Therefore, these columns were dropped using the .drop() function in pandas.

Step 2: Renaming Columns

The column **‘Rocket’** contains the cost of each mission, denoted in Million $.Therefore, the **‘Rocket’** column was renamed to ***‘COM in $ Million’*** to better reflect the cost data it contains.

Step 3: Extracting Countries from the 'Location' Column

The ‘Location’ column contains both the city and country names. To simplify the analysis, we **created a new column ‘Countries’** by extracting the country name from the location information. The extraction process involved:

- Splitting the ‘Location’ column by commas using the .str.split(“,”).str[-1] function to extract the country name.
- Removing any leading or trailing spaces with the .strip() method.

Step 4: Correcting Inaccurate Country Names

Upon reviewing the ‘Countries’ column using the .unique() function, we **found several inaccurate country names.** To rectify this:

We referred to external sources (Wikipedia) to verify the correct names of these countries.

We then created a dictionary called **‘corrected_country_dictionary’** containing the inaccurate and accurate country name pairs.

The .replace() function was used to apply these corrections to the ‘Countries’ column.

Step 5: Extracting Year and Month from the 'Datum' Column

The ‘Datum’ column contains the launch dates in a string format. To extract the year and month of the launch, we **created two new columns ‘Year’** and **‘Month’** which would store the year and the month of launch respectively, using the following approach:

To extract the Year of launch :

- Split the ‘Datum’ column by commas using .str.split(',').
- Retrieved the year information from the appropriate position (index 0) and converted it into an integer using .astype(int).

This new ‘Year’ column will help in further **time-series analysis** of launch events.

To extract the Month of launch:

- Split the ‘Datum’ column by commas using .str.split(',')

- Retrieved the month from the appropriate position (index 1)

Step 6: Extracting Launch Pad Information

The **'Location'** column consists of the names of launch pads used for that particular mission. So, to extract the names of the launch pad we first created a new column named **'Launch Pad'** and then extract the names by using the following approach:

- Split the 'Location' column by commas using .str.split(',')
- Retrieved the name of the launch pad from the appropriate position (index 0)

Step 7: Imputing Missing Values in 'COM in $ Million' Column

When we further inspect the data using the function **'df.info( )'** we get to know that the column **'COM in $ Million'** has multiple missing values, which could affect the data integrity and accuracy of the of the forthcoming analysis.Therefore, to prevent this we replace the missing values by the mean cost for the corresponding Company Name to preserve data integrity and the accuracy of the analysis.

Step 8: Final Check

After completing the cleaning of data by extracting values and and replacing the null values we give a final check to our data to check weather there are anymore columns with missing values in our dataset before we start our EDA. For this final check we use the **'df.isnull( )'** and **'sum( )'** functions, wherein, the **'df.isnull( )'** checks for any null values in the dataset and the **'sum( )'** returns the count of the number of null values present.

By performing this step we find out that there are no more null/missing values present in our dataset and we can now finally start the EDA.

Exploratory Data Analysis:

Number of launches by each company:

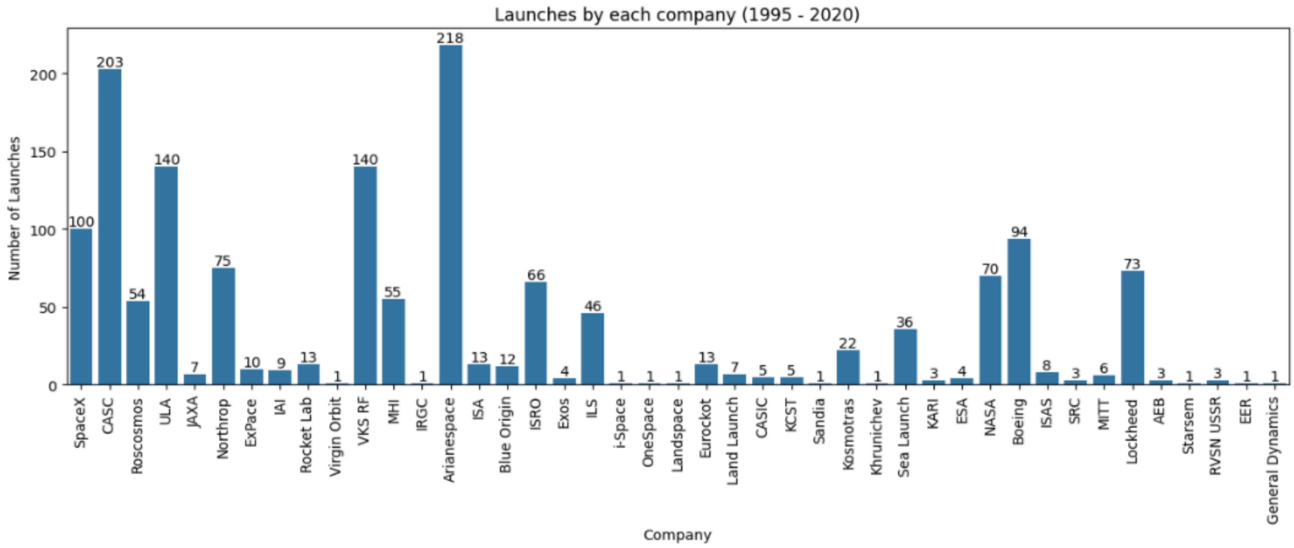
Goal : To analyze the number of launches conducted by each company and identify the leading players in the space exploration industry.

We created a bar plot to visualize the number of space mission launches carried out by each company between 1995 and 2020.Certain companies, such as Arianspace, CACS, VKS RF and ULA stand out as leading players with significantly more launches. This suggests their dominance in space exploration, likely due to their technological capabilities and resources, while companies with fewer launches might be smaller or newer entrants to the industry.This analysis is crucial in identifying the major players in space exploration over the years, helping to **understand which companies have made the most significant contributions.**

Code Snippet:

```
#Company wise plot

plt.subplot(2,2,1)
```

```
company_wise_count=sns.countplot(data=df, x="Company Name")

company_wise_count.bar_label(company_wise_count.containers[0])

plt.xticks(rotation=90)

plt.xlabel("Company")

plt.ylabel("Number of Launches")

plt.title("Launches by each company (1995 - 2020)")
```



Country-wise Launch Count

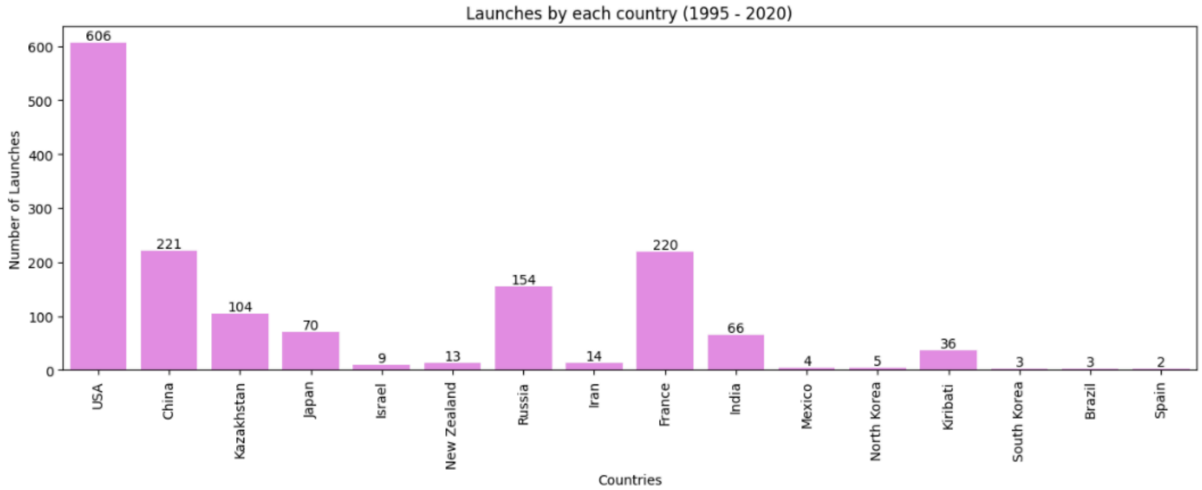
Goal: To determine the distribution of launches across different countries.

This graph shows the distribution of space launches across different countries from 1995 to 2020. The x-axis lists the countries involved in space missions, and the y-axis represents the total number of launches.The **country-wise analysis** shows which nations led in space missions during the same period. Countries like USA, France, China and Russia were at the forefront, with the most launches, indicating their advanced space programs and consistent investments. Meanwhile, countries with fewer launches reflect either growing interest in space exploration or limited resources in conducting space missions.

Code Snippet:

```
#Countrywise plot
plt.subplot(2,2,2)
country_wise_count=sns.countplot(data=df, x="Countries",
color='violet')
```

```
country_wise_count.bar_label(country_wise_count.containers[0])
plt.xticks(rotation=90)
plt.xlabel("Countries")
plt.ylabel("Number of Launches")
plt.title("Launches by each country (1995 - 2020)")
```



Time Series Plot of Launches Per Year

Goal: To analyze the temporal trend of space mission launches over time and identify any notable spikes or patterns.

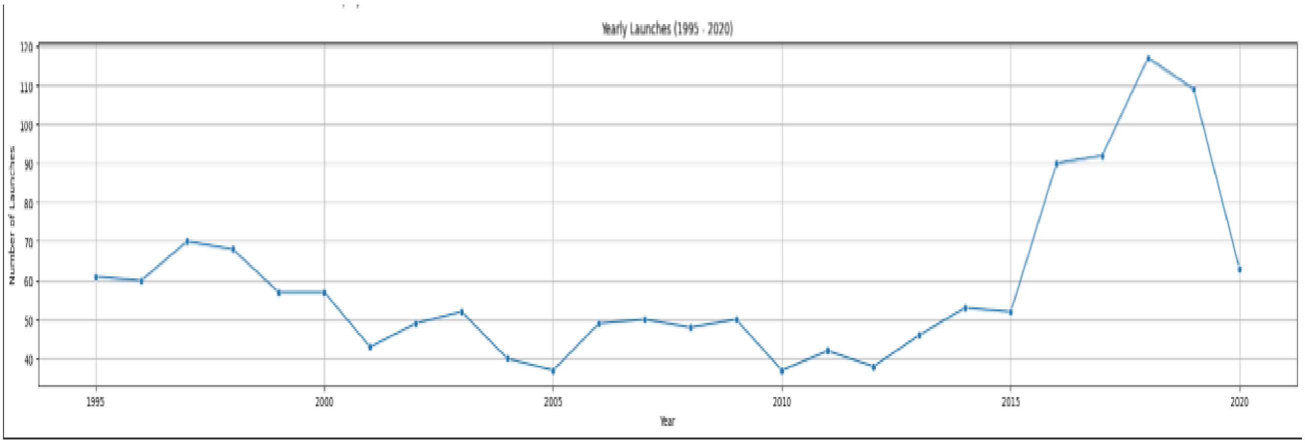
A line plot was used to represent the number of space launches per year from 1995 to 2020. The data was first grouped by the year, and then the total number of launches per year was calculated. The line plot is marked with circular points to highlight individual yearly values, allowing for better visibility of changes over time. A grid was added for clarity, helping to observe trends more easily.

The trend reveals a fluctuating pattern in space mission launches over the years. There was a noticeable increase in launches during the early 2000s, followed by a dip around the mid-2000s and a notable resurgence in launches during the 2010s, largely driven by private companies like SpaceX.Afterward, the number of launches began to increase again, peaking towards 2018-2020. This rise in recent years suggests renewed interest and investment in space exploration, likely driven by private companies and new space ventures.

Code Snippet:

```
#Yearly launches

plt.subplot(2,1,2)

df['Year']=pd.to_numeric(df['Year'], errors='coerce')

df['Year'].dropna()

year_wise_count=df.groupby('Year').size().reset_index(name='Count')
```

```
sns.lineplot(data=year_wise_count, x='Year', y='Count', marker='o')

plt.grid(True)

plt.xlabel('Year')

plt.ylabel('Number of Launches')

plt.title('Yearly Launches (1995 - 2020)')
```



Geographical Distribution of Launches

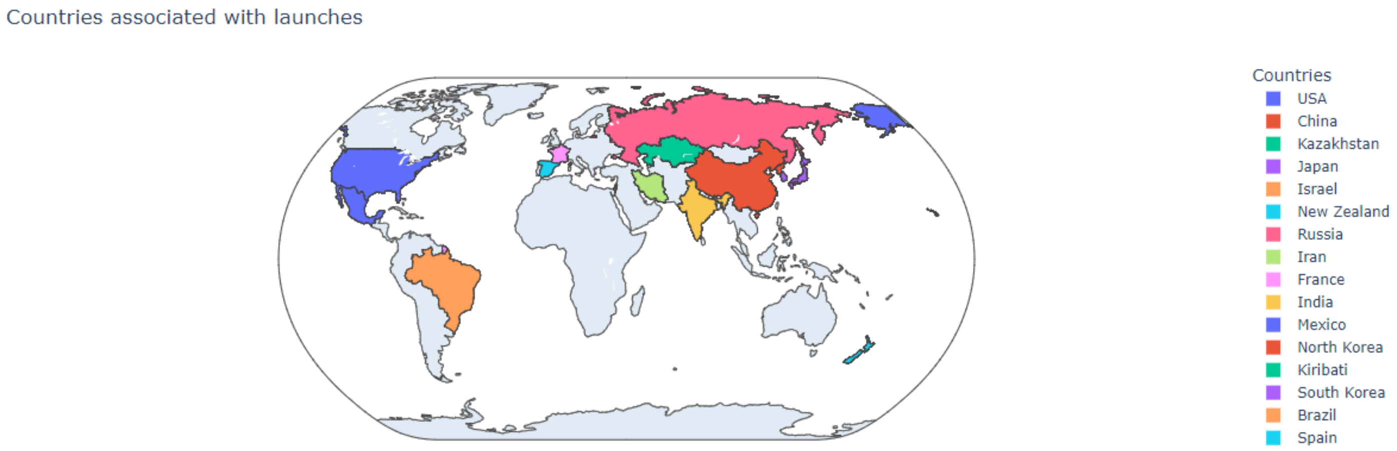
Goal: To visualize the number of space mission launches conducted by each country from 1957 to 2020 and explore the global distribution of space exploration efforts.

A count plot was used to represent the number of space launches by country. The plot shows the total number of launches conducted by each country. Each bar is labeled with the exact count, which makes it easy to observe **which countries have been the most active in space exploration.**

The analysis highlights the dominance of spacefaring nations like the United States, Kazakhstan, and Russia, which have conducted the majority of space missions. Smaller yet notable contributions from other countries are also visible, reflecting the global nature of space exploration over the years. This visualization offers valuable insights into how space missions are distributed across different countries worldwide.

Code Snippet:

```
#Map plot

fig=px.choropleth(df, locations='Countries', color='Countries',
hover_name='Countries', locationmode='country names', projection ='natural earth')

fig.update_layout(title_text='Countries associated with launches')

fig.show()
```



Success Analysis:

Mission success Rate by country

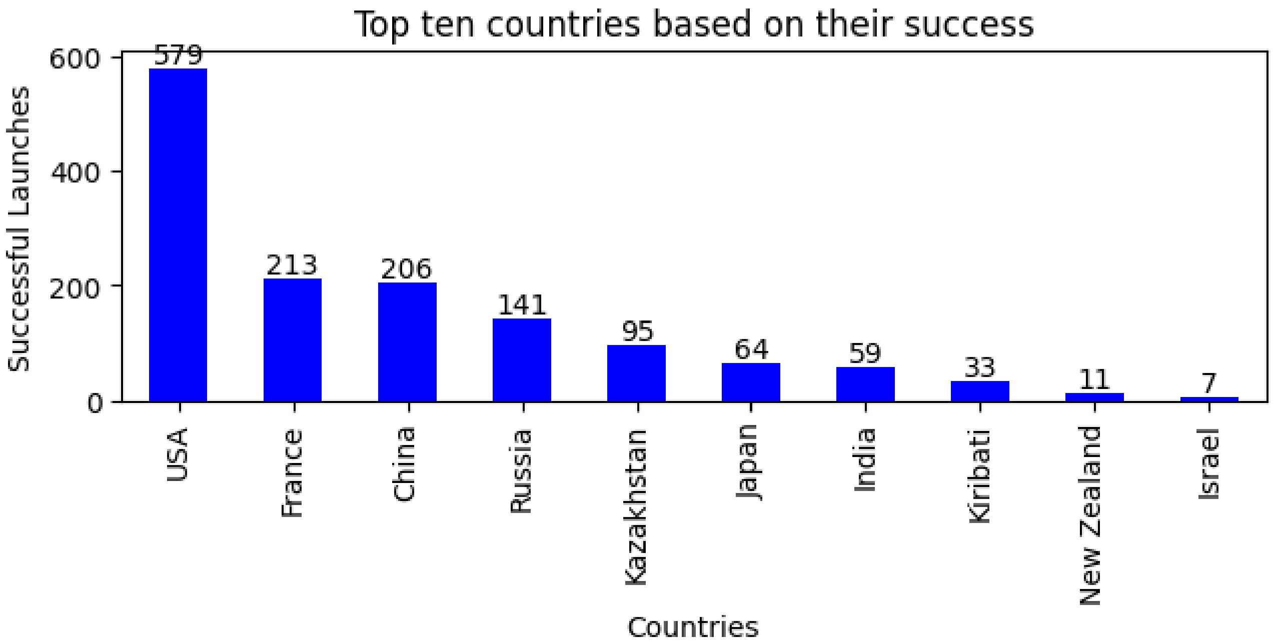
Goal: To evaluate the success rate of space missions by country, identifying which nations have the highest number of successful space launches.

A bar plot was created to showcase the top 10 countries with the highest number of successful space launches. This analysis focuses only on successful missions, filtering the data by mission status. The countries are ranked based on the number of successful launches, and the top 10 countries are displayed. The bars are colored in blue for visual consistency, and each bar is labeled with the exact count of successful launches.

This plot highlights the countries that consistently perform successful space missions,with the United States, Franceand China emerged as the top three countries with the highest number of successful space missions followed by Russia.

Code Snippet:

```
#Top 10 Countries/Success

plt.subplot(4,3,1)

top_ten_countries_success=df[df["Status
Mission"]=="Success"]['Countries'].value_counts().head(10).plot.bar(ylabel='Success
ful Launches', color='blue', title='Top ten countries based on their success')

top_ten_countries_success.bar_label(top_ten_countries_success.containers[0])
```



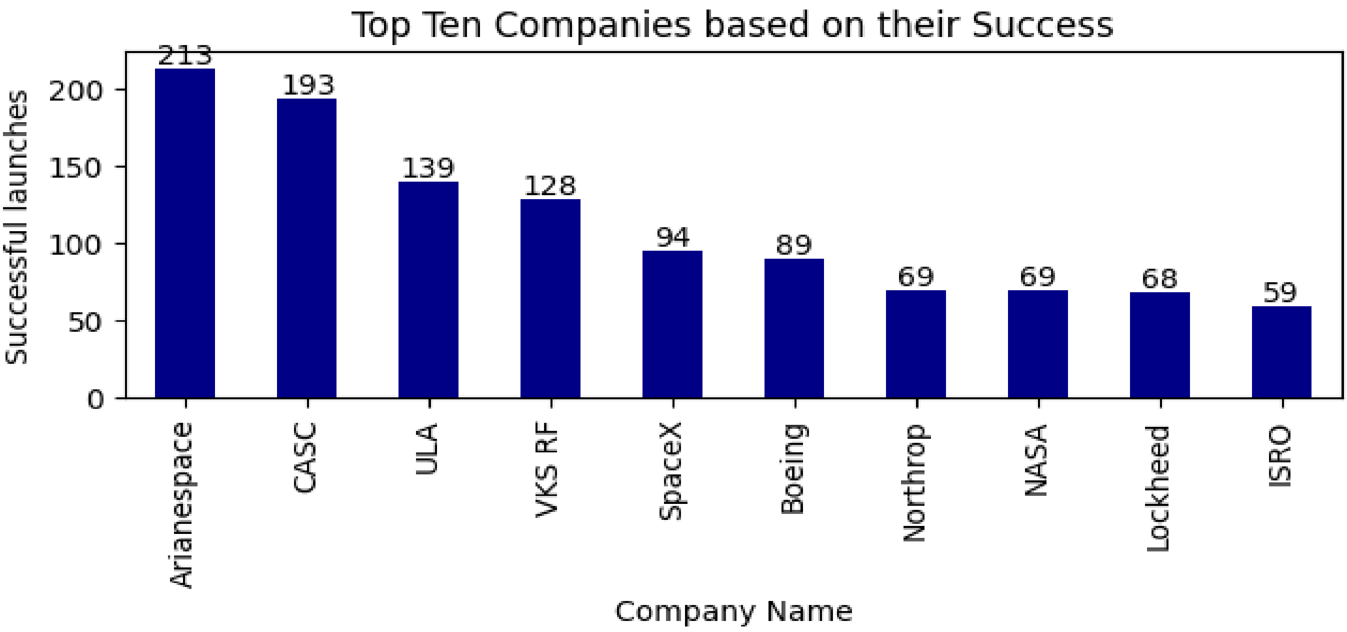
Mission Success Rate by Company

Goal: To evaluate the mission success rates of different companies and assess their operational reliability.

We analyzed the number of successful missions conducted by each company through a bar plot. Certain companies, such as Arianspace, CACS, VKS RF and ULA stand out as leading players with significantly highest success rate in space missions.This analysis helps us understand which organizations consistently perform successful missions, thereby highlighting the top players in space exploration based on mission success.

Code Snippet:

```
#Top 10 Company/Success

plt.subplot(4,3,2)

top_ten_companies_success=df[df['Status Mission']=='Success']['Company
Name'].value_counts().head(10).plot.bar(ylabel='Successful launches',
color='darkblue', title='Top Ten Companies based on their Success')

top_ten_companies_success.bar_label(top_ten_companies_success.containers[0])
```



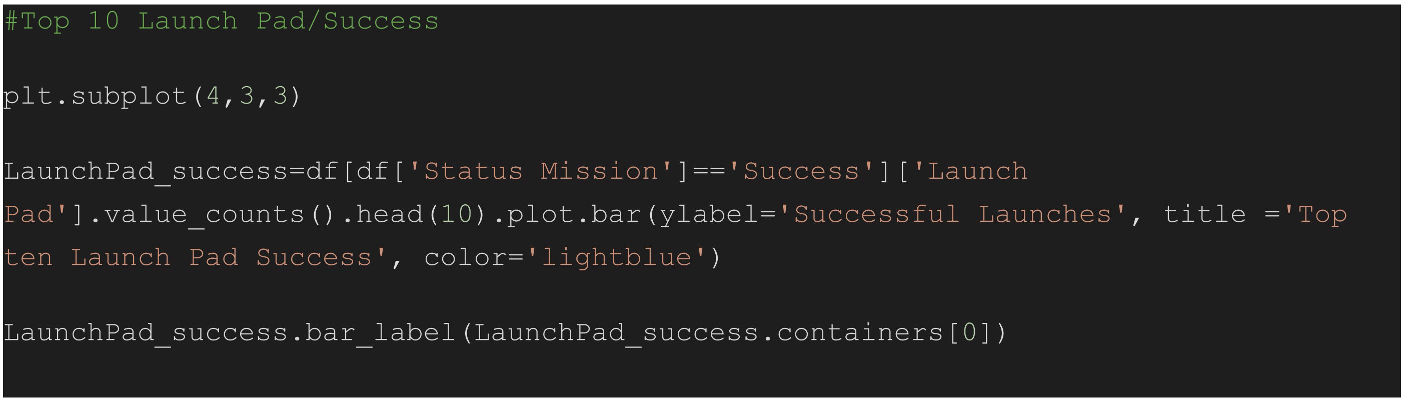
Top 10 Launch Pads by Successful Missions

Goal: To analyze the effectiveness of different launch pads by examining the number of successful space missions associated with each pad

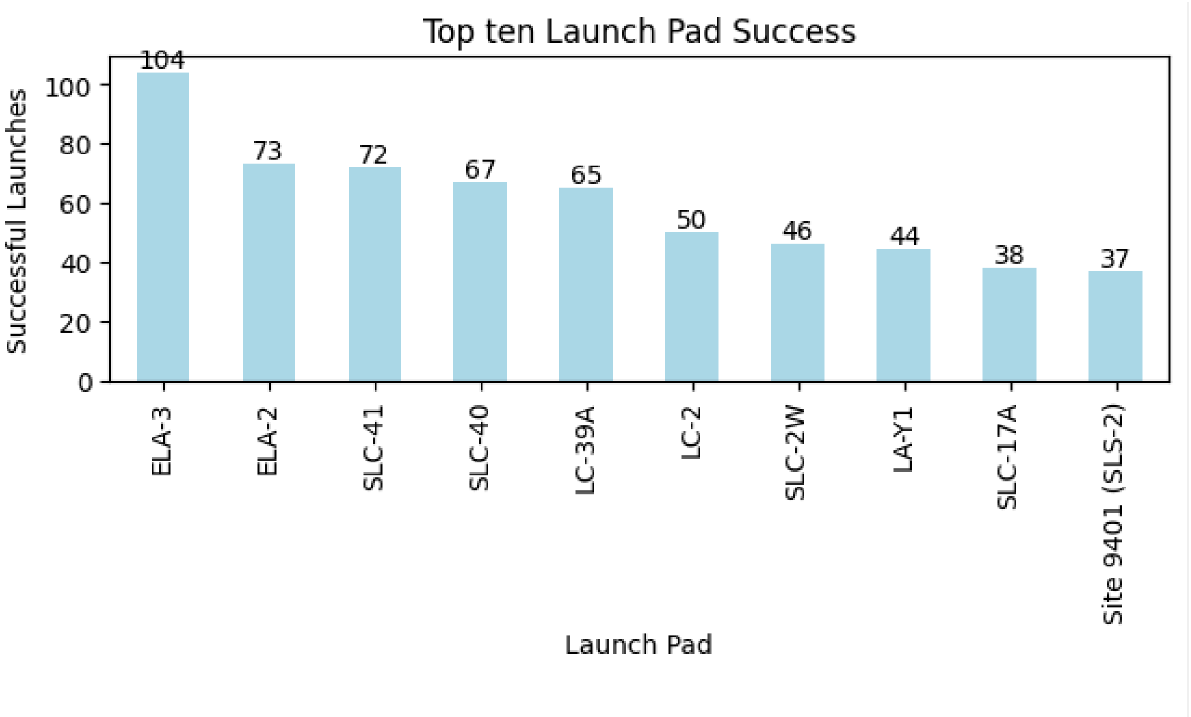
A bar plot was created to showcase the top 10 launch pads with the highest number of successful missions. The analysis filters the dataset to include only successful missions and counts the number of successes for each launch pad.

This analysis highlights the launch pads that have been most successful in facilitating space missions. By identifying which launch pads consistently produce successful outcomes, stakeholders can gain insights into the operational effectiveness and reliability of specific launch sites. This information can be valuable for future planning and investment in space infrastructure, as it underscores the importance of certain launch locations in achieving mission success.

Code Snippet:



```
#Top 10 Launch Pad/Success

plt.subplot(4,3,3)

LaunchPad_success=df[df['Status Mission']=='Success']['Launch
Pad'].value_counts().head(10).plot.bar(ylabel='Successful Launches', title ='Top
ten Launch Pad Success', color='lightblue')

LaunchPad_success.bar_label(LaunchPad_success.containers[0])
```



Yearly Success Analysis of Space Missions

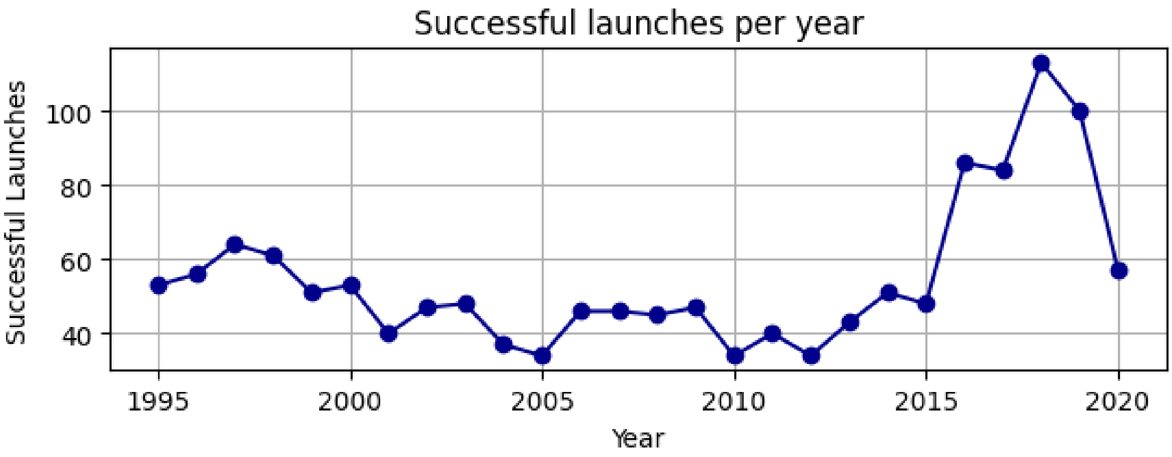
Goal: To analyze the trend of successful space missions over the years, highlighting how the number of successful launches has changed from 1995 to 2020.

A line plot was created to visualize the number of successful launches per year. The analysis filters the dataset to include only successful missions and counts the number of successful launches for each year. The data is then sorted chronologically, and a line plot is generated with markers to indicate individual yearly values.

This visualization reveals trends and patterns in successful space missions over time. It allows us to observe peaks in successful launches during specific years, reflecting significant advancements in space exploration. The analysis can highlight key historical events or developments in the space industry that may have influenced the number of successful missions, such as the rise of private space companies or technological advancements.

Code Snippet:

```
#Yearly Success

plt.subplot(4,3,4)

df[df['Status
Mission']=='Success']['Year'].value_counts().sort_index().plot.line(ylabel='S
uccessful Launches', title='Successful launches per year',
marker='o',grid=True, color='darkblue')
```



Yearly Success and Cost Analysis of Space Missions

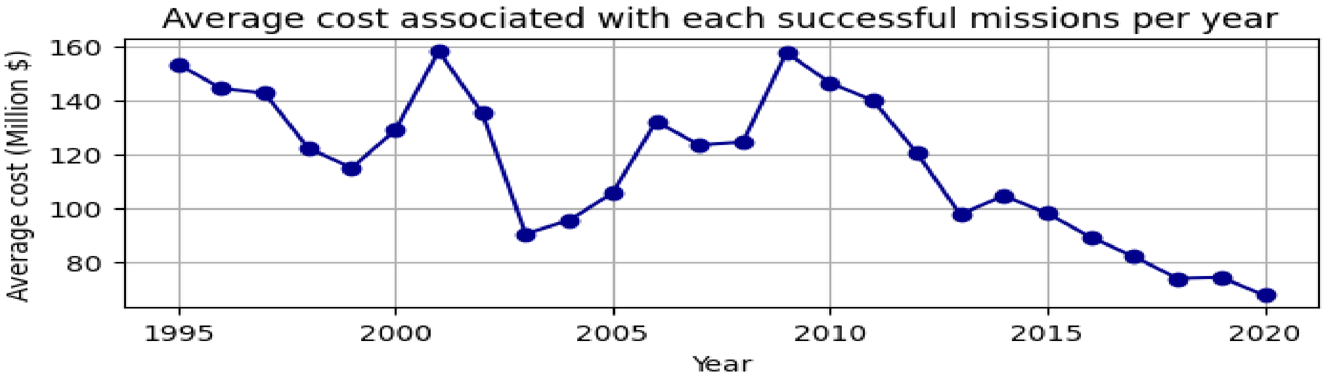
Goal: To examine the average cost associated with successful space missions over the years, highlighting trends in expenditure relative to mission success.

A line plot was created to visualize the average cost (in million dollars) of successful missions for each year. The analysis filters the dataset to include only successful missions and calculates the mean cost for each year. This data is then plotted as a line graph, with markers indicating the average cost for individual years.

This visualization illustrates how the average expenditure on successful missions has evolved over time. Notably, the trend may indicate a reduction in costs due to advancements in technology, improved launch processes, and increased competition within the space industry. By observing the average costs alongside the trends in successful missions, we can infer how innovations and efficiencies have contributed to making space exploration more financially accessible. This analysis provides valuable insights into the economic aspects of space missions and the impact of technological advancements on mission costs.

Code Snippet:

```
#Yearly Success and cost

plt.subplot(4,3,5)

success_missions_cost_and_year=df[df['Status Mission']=="Success"]

success_missions_cost_and_year.groupby('Year')['COM in $
Million'].mean().plot.line(ylabel='Average cost (Million $)', title='Average cost
associated with each successful missions per year', marker='o', grid=True,
color='darkblue')
```



Expenditure Analysis of Successful Missions by Country

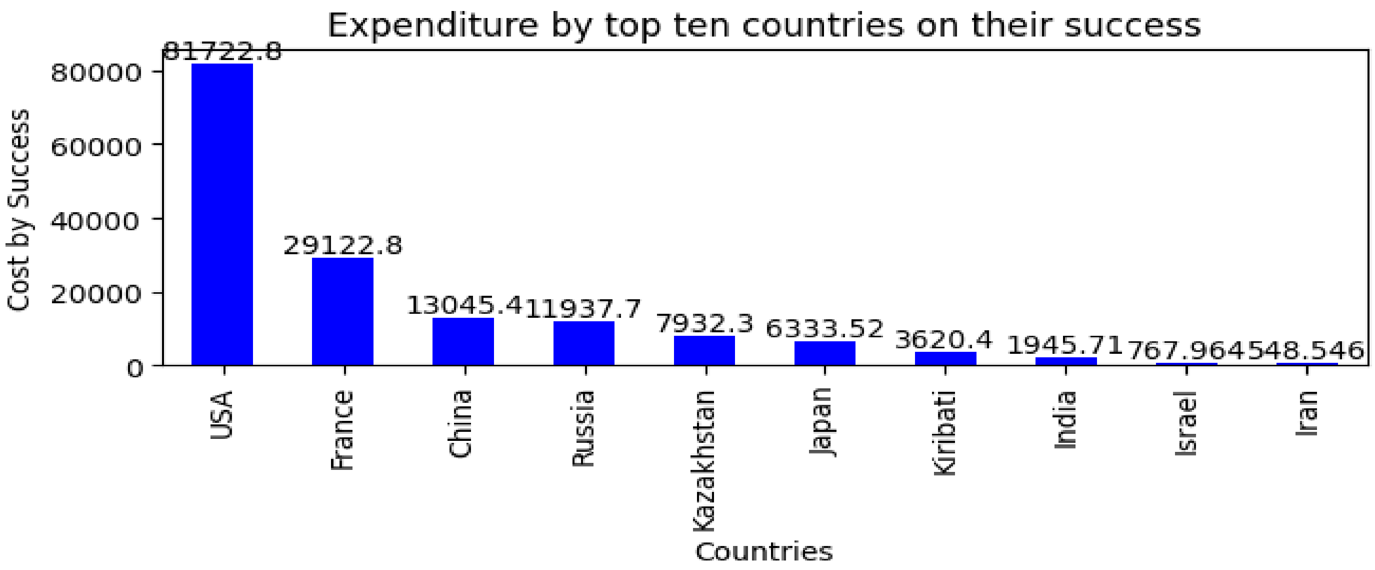
Goal: To analyze the total expenditure on successful space missions by the top 10 countries, highlighting their financial commitment to space exploration.

A bar plot was created to display the total cost (in million dollars) associated with successful missions for the top 10 countries. The analysis filters the dataset to include only successful missions and groups the total costs by country. The countries are then ranked based on their total expenditure, with the highest spenders displayed in the plot.

This analysis provides insights into how much each of the top-performing countries invests in their successful space missions. It emphasizes the financial resources allocated to achieving mission success and illustrates the commitment of various nations to advancing their space programs. Countries that typically invest heavily in space exploration, such as the United States and Russia are likely to feature prominently in this analysis, showcasing their expenditures in relation to mission success.

Code Snippet:

```
#Top 10 Countries/Success/Cost

plt.subplot(4,3,7)

country_success=df[df['Status Mission']=='Success']['Countries']

success_missions_cost_and_country=df['COM in $
Million'].groupby(country_success).sum().sort_values(ascending=False).head(10).plot
.bar(ylabel='Cost by Success', title='Expenditure by top ten countries on their
success',color='blue')

success_missions_cost_and_country.bar_label(success_missions_cost_and_country.conta
iners[0])
```



Expenditure Analysis of Successful Missions by Company

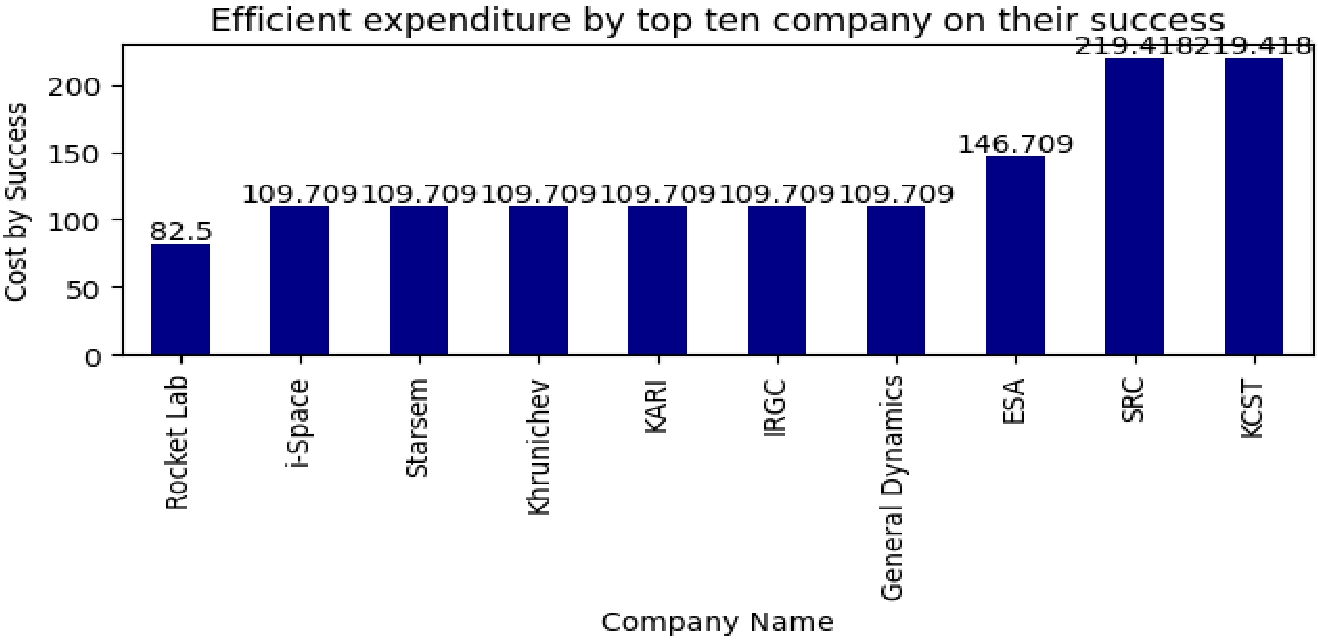
Goal: To evaluate the total expenditure on successful space missions by the top 10 companies, highlighting their financial investment in achieving mission success.

A bar plot was created to display the total cost (in million dollars) associated with successful missions for the top 10 companies. The analysis filters the dataset to include only successful missions and groups the total costs by company. Companies are ranked based on their total expenditure, with the highest spenders showcased in the plot.

This analysis provides insights into how much each of the leading companies invests in their successful space missions. It highlights the financial resources dedicated to achieving success in space exploration and illustrates the operational scale of various organizations.

Code Snippet:

```
#Top 10 Company/Success/Cost

plt.subplot(4,3,8)

company_success=df[df['Status Mission']=='Success']['Company Name']

success_missions_cost_and_company=df['COM in $ Million'].groupby(company_success).sum().sort_values(ascending=False).head(10).plot.bar(ylabel='Cost by Success', title='Expenditure by top ten company on their success',color='darkblue')

success_missions_cost_and_company.bar_label(success_missions_cost_and_company.containers[0])
```



Cost Efficiency of Successful Missions by Country

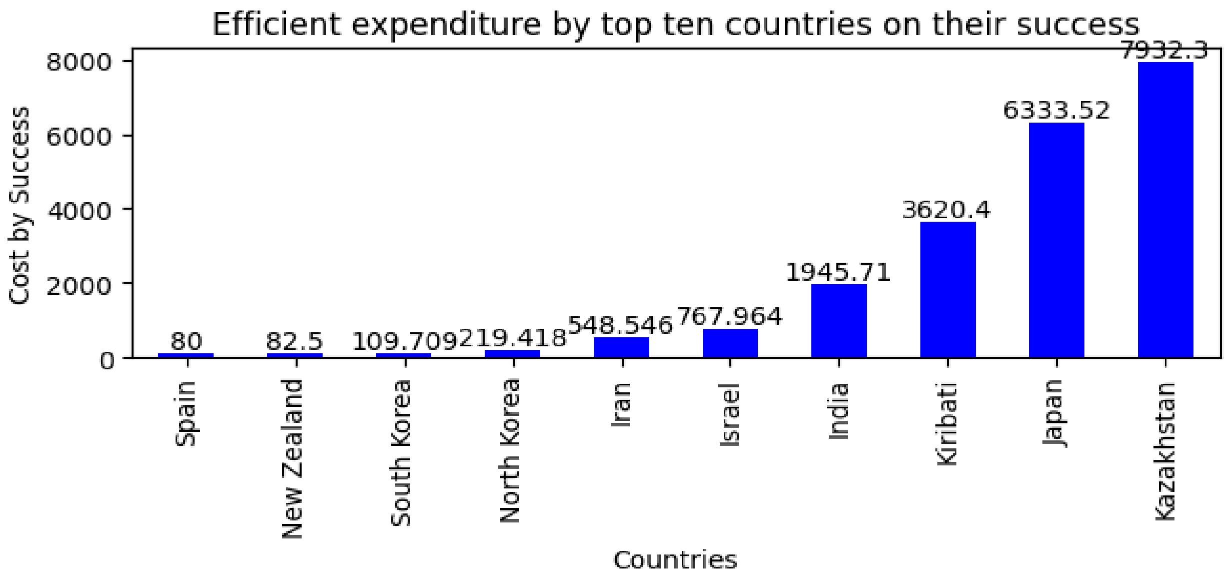
Goal: To analyze the cost efficiency of successful space missions by country, identifying the countries that have achieved the most success with the least expenditure.

A bar plot was created to showcase the top 10 countries that have spent the least amount on their successful space missions. The analysis calculates the total mission cost (in million dollars) associated with successful launches for each country. The countries are then ranked based on the total cost, and the top 10 countries with the most cost-efficient success are displayed.

This analysis highlights the countries that have achieved a significant number of successful missions while keeping the costs relatively low. It provides insight into which nations are most efficient in terms of resource usage, reflecting a strategic and economical approach to space exploration.

Code Snippet:

```
#Top 10 efficient Country/Success/Cost

plt.subplot(4,3,10)

country_success=df[df['Status Mission']=='Success']['Countries']

cost_efficient_country_success=df['COM in $ Million'].groupby(country_success).sum().sort_values(ascending=True).head(10).plot.bar(ylabel='Cost by Success', title='Efficient expenditure by top ten countries on their success',color='blue')

cost_efficient_country_success.bar_label(cost_efficient_country_success.containers[0])
```

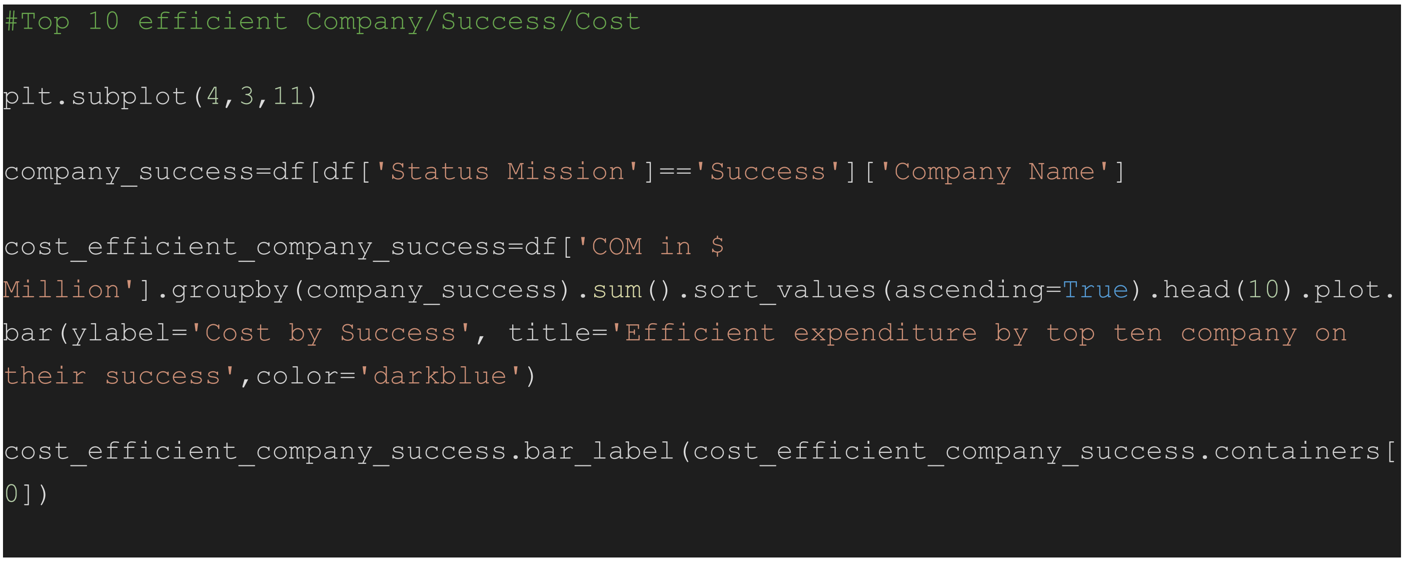


Cost Efficiency of Successful Missions by Company

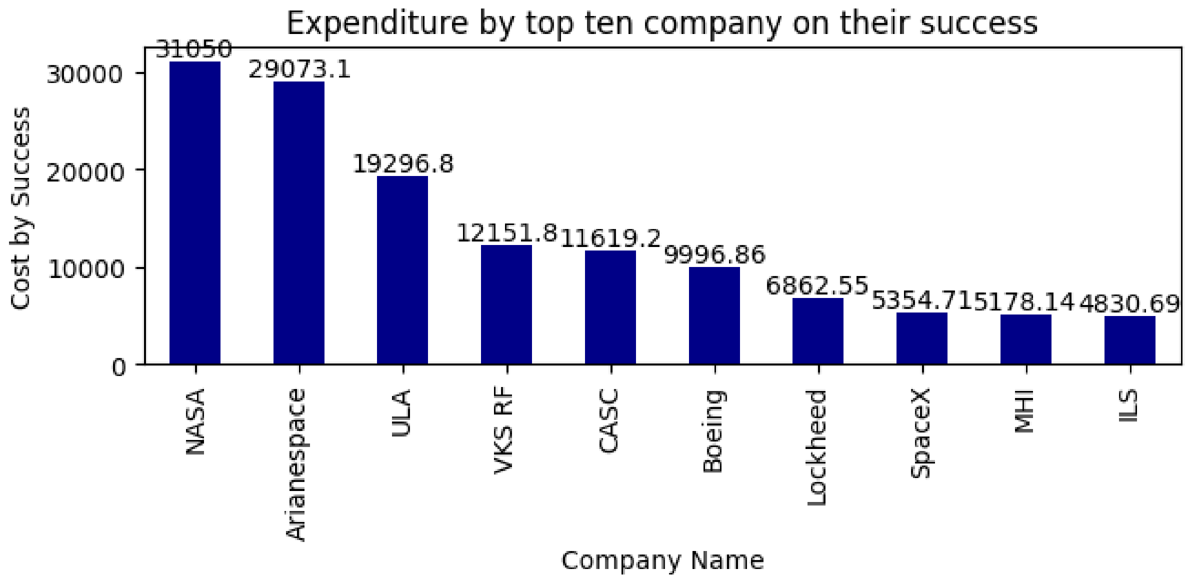
Goal: To assess the cost efficiency of successful space missions conducted by various companies, identifying those that have achieved the most success with minimal expenditure.

A bar plot was created to display the top 10 companies that have the most cost-effective successful missions. The analysis calculates the total cost (in million dollars) associated with successful missions for each company. Companies are ranked based on their total expenditure on successful launches, and the top 10 are highlighted in the plot.This analysis reveals which companies have managed to conduct successful missions while spending the least amount of money. It provides insights into the operational effectiveness and financial management of various space companies.

Code Snippet:



```
#Top 10 efficient Company/Success/Cost

plt.subplot(4,3,11)

company_success=df[df['Status Mission']=='Success']['Company Name']

cost_efficient_company_success=df['COM in $
Million'].groupby(company_success).sum().sort_values(ascending=True).head(10).plot.
bar(ylabel='Cost by Success', title='Efficient expenditure by top ten company on
their success',color='darkblue')

cost_efficient_company_success.bar_label(cost_efficient_company_success.containers[
0])
```



Failure Analysis:

Top 10 Countries by Failed Missions

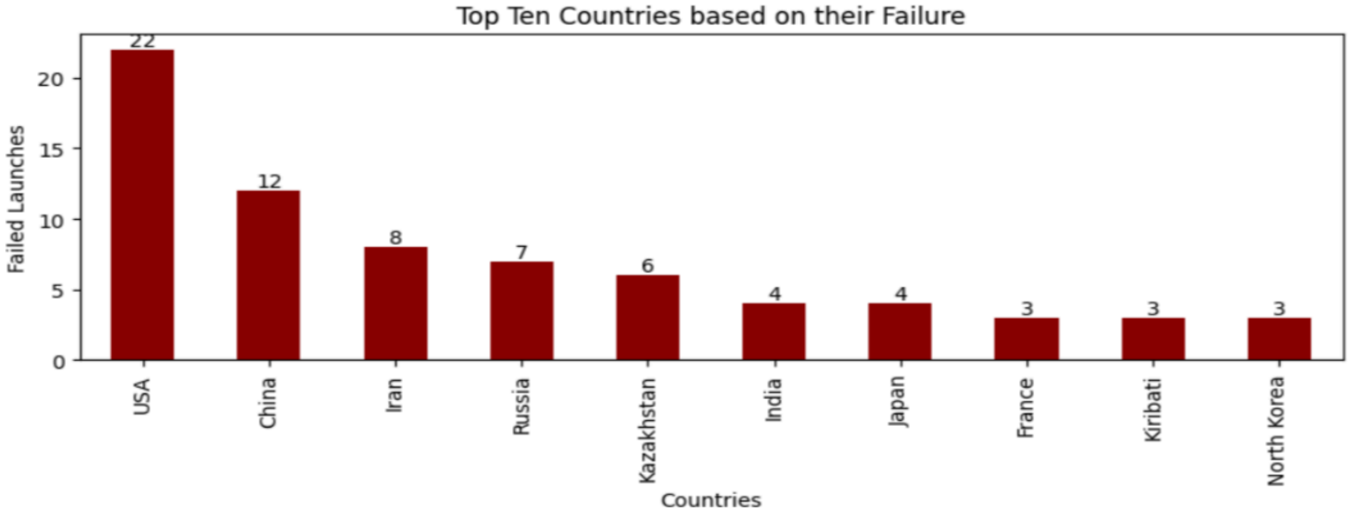
Goal: To analyze the number of failed space missions by country, identifying which nations have experienced the most failures in their space exploration efforts.

A bar plot was created to display the top 10 countries with the highest number of failed space missions. The analysis filters the dataset to include only missions that resulted in failure and counts the number of failures for each country.

By examining the countries with the most failed missions, we can identify potential issues related to technology, planning, or execution. Understanding these failures can be crucial for improving future missions and learning from past mistakes, thereby enhancing the overall success rate in space exploration.

Code Snippet:

```
plt.subplot(3,2,1)

top_ten_countries_failure = df[df['Status
Mission']=='Failure']['Countries'].value_counts().head(10).plot.bar(ylabel= 'Failed
Launches', title='Top Ten Countries based on their Failure', color='darkred')

top_ten_countries_failure.bar_label(top_ten_countries_failure.containers[0])
```



Top 10 Companies by Failed Missions

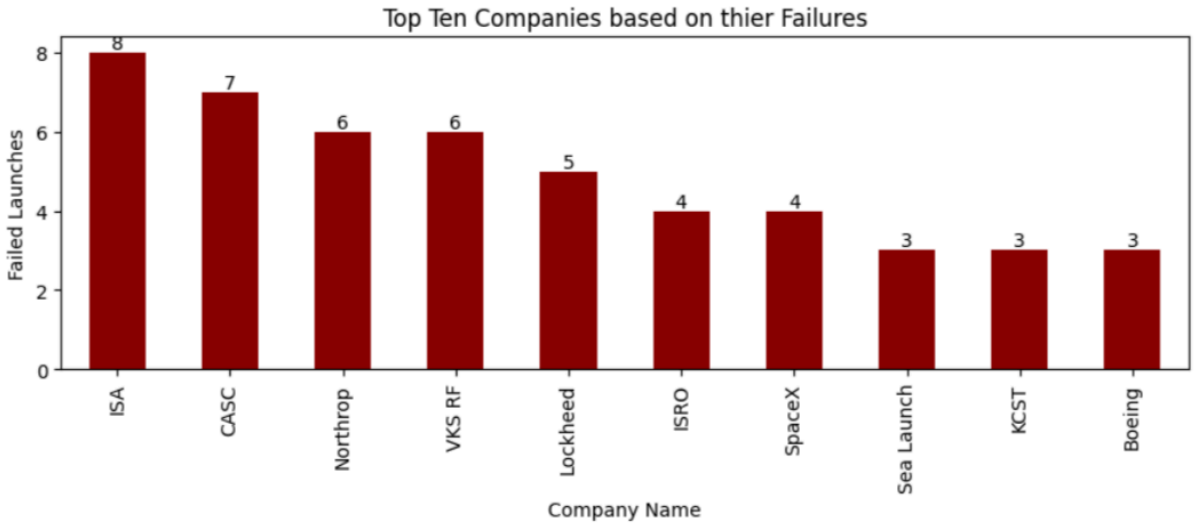
Goal: To analyze the number of failed space missions by company, identifying which organizations have faced the most challenges in their space endeavors.

A bar plot was created to display the top 10 companies with the highest number of failed space missions. The analysis filters the dataset to include only missions that resulted in failure and counts the number of failures for each company.

By examining the organizations with the most failures, we can identify trends or common challenges that might affect mission outcomes. Understanding these failures can help companies improve their processes and technologies, ultimately contributing to higher success rates in future missions.

Code Snippet:

```
plt.subplot(3,2,2)

top_ten_companies_failure = df[df['Status Mission']=='Failure']['Company
Name'].value_counts().head(10).plot.bar(ylabel='Failed Launches',title='Top Ten
Companies based on thier Failures',color='darkred')

top_ten_companies_failure.bar_label(top_ten_companies_failure.containers[0])
```



Yearly Failure Analysis of Space Missions

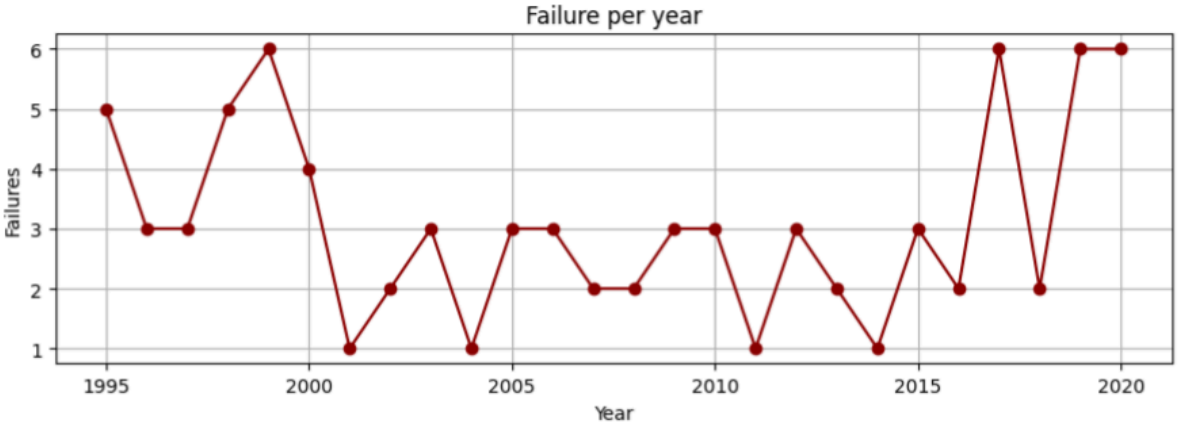
Goal: To examine the trend of failed space missions over the years, highlighting how the number of failures has varied from year to year.

A line plot was created to visualize the number of failed launches per year. The analysis filters the dataset to include only missions that resulted in failure and counts the number of failures for each year. This data is sorted chronologically and plotted as a line graph, with markers indicating the failure count for individual years.

This visualization illustrates trends in mission failures over time, allowing us to observe peaks in failure rates during specific years. By analyzing this data, we can identify potential factors that may have contributed to increased failures, such as technological challenges, changes in mission objectives, or external influences.

Code Snippet:

```
plt.subplot(3,2,3)

df[df['StatusMission']=='Failure']['Year'].value_counts().sort_index().plot.line(ylabel='Failures',color='darkred', title='Failure per year', marker='o', grid=True)
```



Average Cost Analysis of Different Types of Failures

Goal: To analyze and compare the average costs associated with three distinct types of mission failures: overall failures, partial failures, and prelaunch failures.

A line plot was created to visualize the average cost (in million dollars) associated with each type of failure over the years. The analysis segments the dataset into three categories:

- **Failed Missions**: Missions that did not succeed.
- **Partial Failures**: Missions that experienced some level of success but did not achieve their primary objectives.
- **Prelaunch Failures**: Missions that failed before launch.

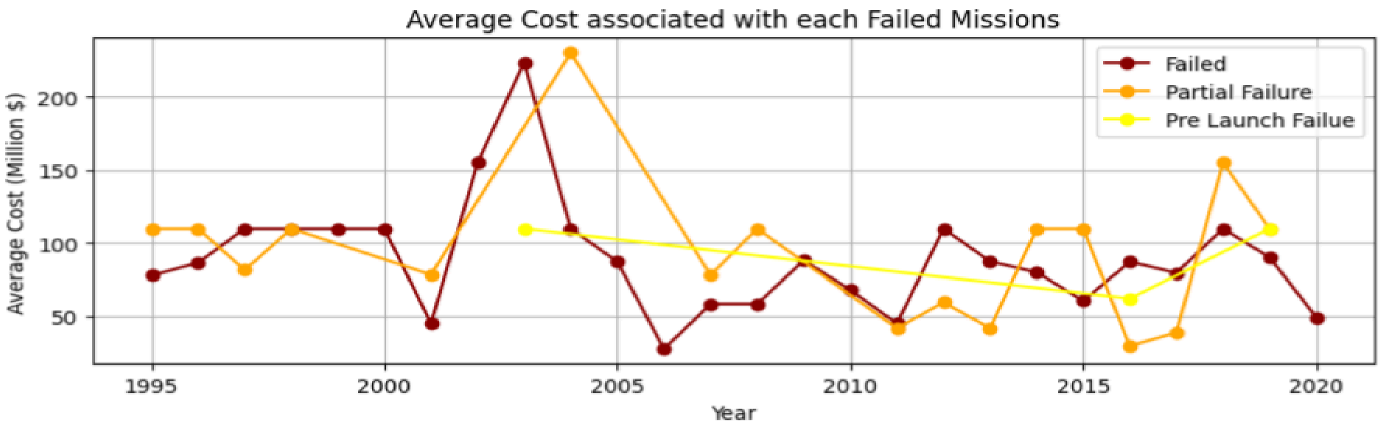
Each type of failure is represented by a different colored line: dark red for overall failures, orange for partial failures, and yellow for prelaunch failures. The average costs for each type are calculated and plotted over the years, with markers indicating individual yearly averages. A legend is included for clarity, and a grid enhances the readability of the plot.

This visualization provides insights into how the costs associated with each type of failure have evolved over time. By comparing the average costs, we can assess which types of failures tend to be more financially burdensome and identify trends that may indicate the need for improved planning or technology to mitigate these costs in future missions.

Code Snippet:

```
plt.subplot(3,2,4)

failed_mission=df[df['Status Mission']=='Failure']

partially_failed_mission=df[df['Status Mission']=='Partial Failure']

prelaunch_failed_mission=df[df['Status Mission']=='Prelaunch Failure']

failed_mission.groupby('Year')['COM in $ Million'].mean().plot.line(marker='o',
grid=True, color='darkred', label='Failed')
```

```
partially_failed_mission.groupby('Year')['COM in $ Million'].mean().plot.line(
marker='o', grid=True, color='orange', label='Partial Failure')

prelaunch_failed_mission.groupby('Year')['COM in $ Million'].mean().plot.line(
marker='o', grid=True, color='yellow', label='Pre Launch Failue')

plt.legend()

plt.title('Average Cost associated with each Failed Missions')

plt.ylabel('Average Cost (Million $)')
```



Cost Analysis of Failed Missions by Top 10 Countries

Goal: To examine the total expenditure on failed space missions for the top 10 countries, highlighting their financial losses in unsuccessful endeavors.

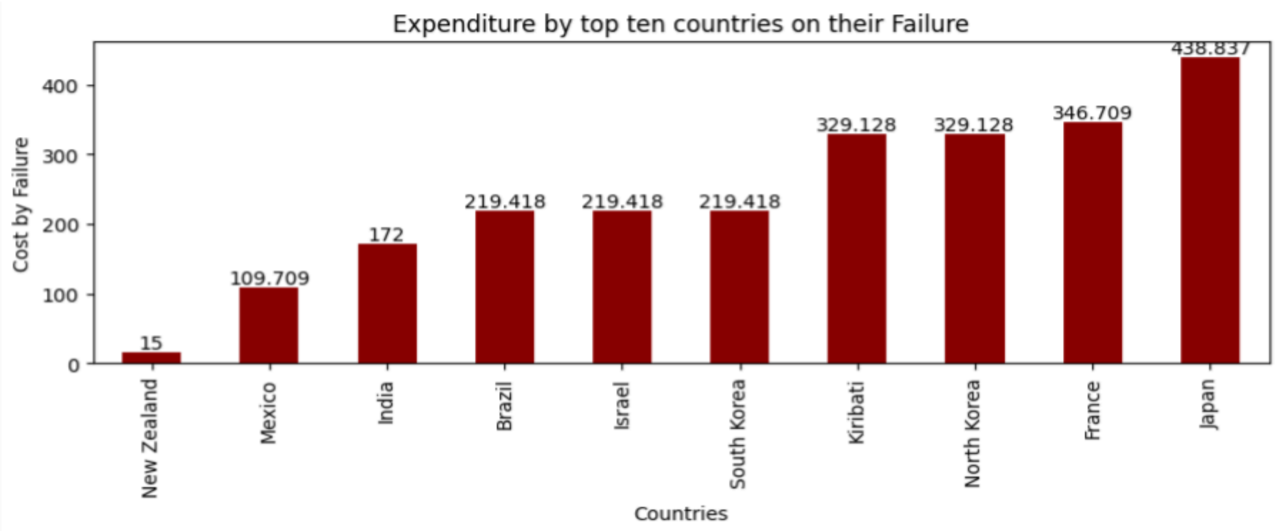
A bar plot was created to display the total cost (in million dollars) associated with failed missions for the top 10 countries. The analysis filters the dataset to include only failed missions and aggregates the costs by country. The countries with the highest expenditures on failed missions are identified and represented in the plot, with bars colored dark red to signify financial losses.

This analysis provides valuable insights into how much each of the leading countries has invested in unsuccessful space missions. It highlights the financial risks associated with space exploration and emphasizes the importance of effective planning and execution. By understanding the costs related to failures, countries can better assess their space programs' efficiency and make informed decisions to minimize future losses and improve overall mission success rates.

Code Snippet:

```
plt.subplot(3,2,5)

failed_mission_country=df[df['Status Mission']=='Failure']['Countries']
```

```
failure_mission_cost_and_company=df['COM in $ Million'].groupby(failed_mission_country).sum().sort_values().head(10).plot.bar(ylabel='Cost by Failure', title='Expenditure by top ten countries on their Failure',color='darkred')

failure_mission_cost_and_company.bar_label(failure_mission_cost_and_company.containers[0])
```



Feature Engineering:

Success Rate Analysis:

Goal: To calculate the success rate of space missions by company and country,highlighting the top performers in terms of both efficiency and experience.

The success rate was calculated using the formula:

Success Rate = Number of Successful Missions / Total Number of Mission

To calculate the Success Rate using this formula,we first grouped the dataset by both **'Company Name'** and '**Country'** to calculate the success rate for each combination and then used a **'Lambda'** function that counts the number of successful missions (i.e., those labelled as 'Success') and dividing by the total number of missions for each group. Then stored these success rates of only those Companies and Countries who has carried out more than 50 missions in a column named **'Success Rate'**.

Code Snippet:

```
#Formula : Success Rate= Number of successful missions/Total number of missions

# Grouping by Company and Country, then calculating the success rate

success_rate_df = df.groupby(['Company Name', 'Countries']).apply(

    lambda x: (x['Status Mission'] == 'Success').sum() / len(x)

).reset_index(name='Success Rate')




mission_count_df=df.groupby(['CompanyName','Countries']).size().reset_index(name='Number of Missions')

merged_df = pd.merge(success_rate_df, mission_count_df)

success_rate = merged_df[merged_df['Number of Missions'] > 50].sort_values(by='Success Rate', ascending=False).head(15)

success_rate
```

Cost Efficiency Analysis:

Goal: To evaluate the cost efficiency of space missions by company and country, focusing on how effectively resources are utilized in successful missions.

Cost efficiency was calculated using the formula:

Cost Efficiency = Number of Successful Missions / Total Cost of All Missions

To calculate the Cost Efficiency using this formula,we first grouped the dataset by both **‘Company Name’** and ‘**Country’** to calculate the success rate for each combination and then used a **‘Lambda’** function that counts the number of successful missions (i.e., those labelled as 'Success') and dividing y the total cost of missions present in the column **‘COM in $ Million’**, and then stored the resulted values in a column named **‘Cost Efficiency’.** Additionally, the total number of missions for each company and country was calculated.To ensure a comprehensive view the cost efficiency data was then merged with the mission count. To focus on statistically significant results,the analysis concentrated on entities with more than 50 missions.

Code Snippet:

```
#Formula : Cost efficiency=Number of Successful Missions/ Total cost of all missions
```

```
cost_efficiencty_df=df.groupby(['Company Name','Countries']).apply(

    lambda x:(x['Status Mission']=='Success').sum()/x['COM in $
Million'].sum()

).reset_index(name='Cost Efficiency')



cost_mission_count=df.groupby(['Company
Name','Countries']).size().reset_index(name='Number of Missions')



cost_merged_df=pd.merge(cost_efficiencty_df, cost_mission_count, on=['Company
Name', 'Countries'])



cost_efficiency=cost_merged_df[cost_merged_df['Number of
Missions']>50].sort_values(by='Cost Efficiency', ascending=False).head(15)

cost_efficiency
```

Overall Efficiency Calculation:

Goal: To compute both the unweighted and weighted overall efficiency for space missions, combining **Success Rate** and **Cost Efficiency** metrics.

Overall efficiency was calculated using the formula:

Overall Efficiency = Success Rate+Cost Efficiency / 2

Overall Efficiency along with weights was calculated using the formula:

Overall Efficiency (Weighted)=(w1×Success Rate)+(w2×Cost Efficiency)

Where, w1=0.7and w2=0.3

We calculate the Overall Efficiency to reflect both the success rate and cost efficiency of space missions. This helps evaluate which companies and countries are not only successful in their missions but also financially efficient.

Here, while calculating the Overall Efficiency we have introduced weights to both the factors considered in our calculation i.e. we assigned a weight **‘w1’** as **0.7** to **‘Success Rate’** indicating more focus on the success of the mission since success of a mission is more important than the cost it incurs and assign a weight **‘w2’** as **0.3** to **‘Cost Efficiency’**.

Code Snippet:

```
#Formula : Overall Efficency = (Success Rate + Cost Efficiency)/2

efficiency = pd.merge(success_rate, cost_efficiency)

efficiency

efficiency['Overall Efficiency']=(efficiency['Success Rate']+efficiency['Cost
Efficiency'])/2

w1, w2 = 0.7, 0.3

efficiency['Overall Efficiency (Weighted)'] = (w1 * efficiency['Success
Rate']) + (w2 * efficiency['Cost Efficiency'])

efficiency
```

Model Implementation:

Regression:

Regression is a type of analysis used to understand the relationship between variables. In simple terms, it helps predict a certain value (like sales, temperature, or costs) based on other related data. It comes under **Supervised Machine Learning.**

For example, if you want to predict the cost of a car based on its age and mileage, regression would allow you to find a pattern and make accurate predictions.

In essence, regression helps us estimate or forecast a continuous outcome (like a number) based on other known data points.

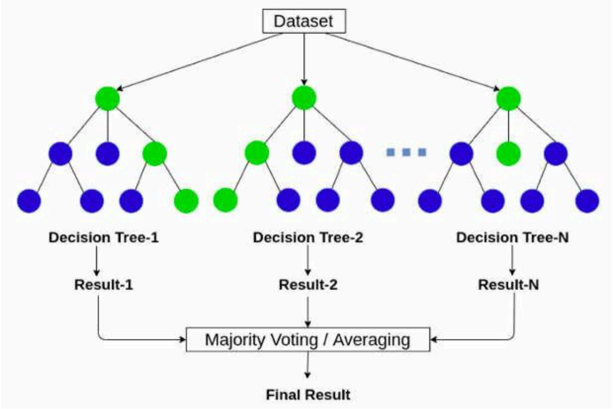
Ensemble learning technique:

Ensemble learning technique is technique of asking multiple experts instead of relying on single opinion. It combines the predictions of several models to make a final decision or prediction, which usually results in more accurate and reliable outcomes. The idea is that by using a group of models (like a "team of experts"), we can reduce the errors that a single model might make.

Model 1: Random Forest Regressor

RandomForest Regressor is a powerful ensemble learning technique that is used for regression tasks.

It builds multiple decision trees during training and merges the output to improve the accuracy and control the overfitting.



RandomForest regressor uses bootstrapping, a sampling technique where multiple subsets of training data are created by sampling replacement. Each tree is trained on different subset.

Advantages of RandomForest:

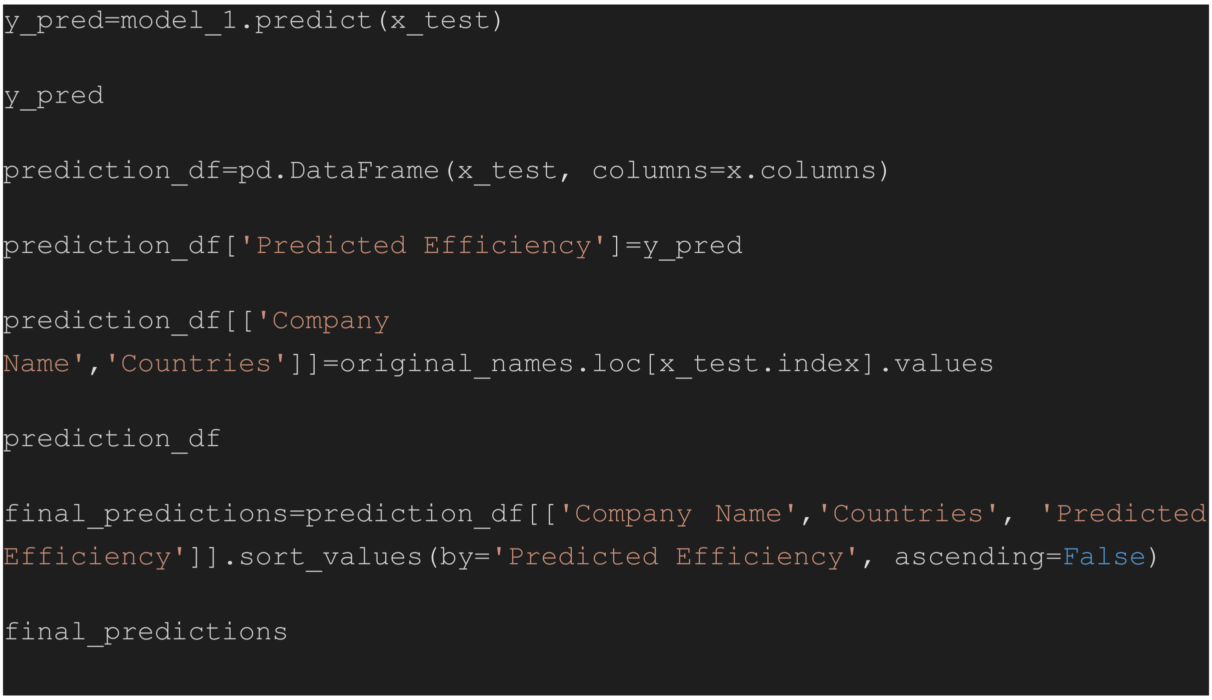
1) Less sensitive to overfitting
2) Can capture complex relationships in data
3) Can provide insights into the importance of different features.

Using Randomforest Regressor we got an accuracy of 72% approx, where the predicted countries and companies are:

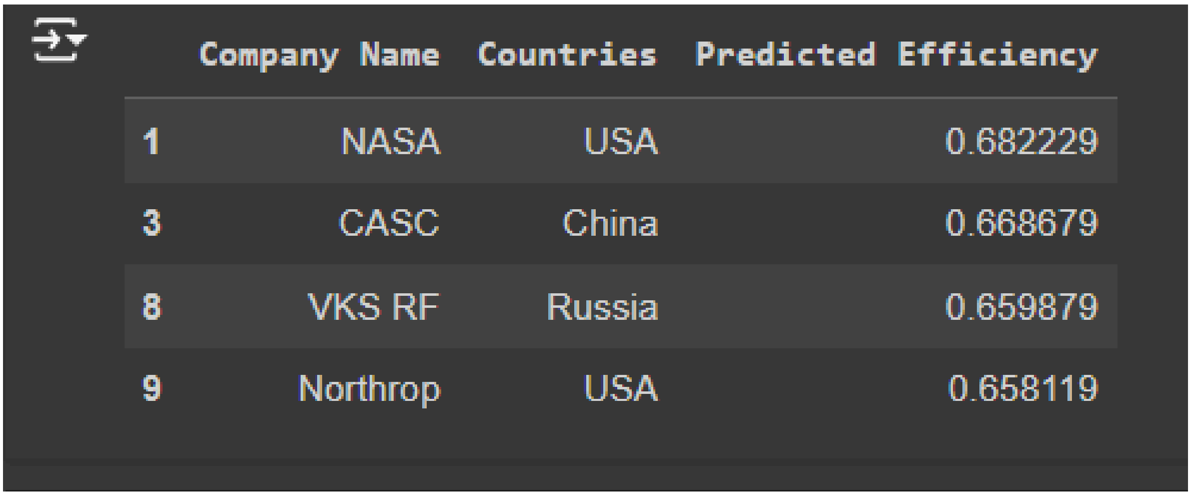
1) **NASA - USA**
2) **CASC - China**
3) **VKS RF - Russia**
4) **Northrop - USA**

Code Snippet:

```
x=pd.get_dummies(efficiency[['Company Name', 'Countries', 'Number of Missions', 'Success Rate', 'Cost Efficiency']], drop_first=True)

y=efficiency['Overall Efficiency (Weighted)']

original_names=efficiency[['Company Name','Countries']]

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=52)

model_1=RandomForestRegressor(n_estimators=100, random_state=42)

model_1

model_1.fit(x_train, y_train)
```



```
y_pred=model_1.predict(x_test)

y_pred

prediction_df=pd.DataFrame(x_test, columns=x.columns)

prediction_df['Predicted Efficiency']=y_pred

prediction_df[['Company
Name','Countries']]=original_names.loc[x_test.index].values

prediction_df

final_predictions=prediction_df[['Company Name','Countries', 'Predicted
Efficiency']].sort_values(by='Predicted Efficiency', ascending=False)

final_predictions
```



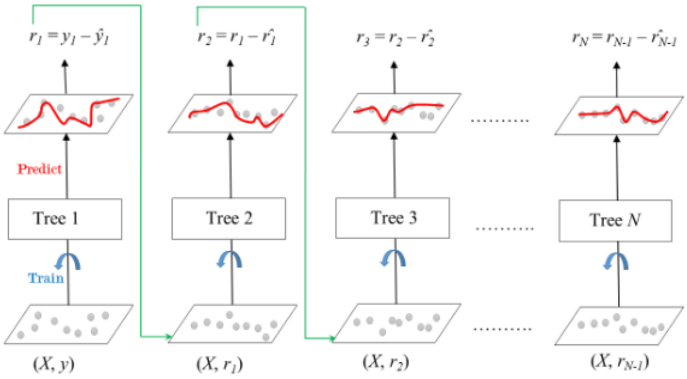
```
     Company Name  Countries  Predicted Efficiency
1            NASA        USA              0.682229
3            CASC      China              0.668679
8          VKS RF     Russia              0.659879
9        Northrop        USA              0.658119
```

Model 2 : Gradient Boosting Regressor

It is a popular Machine Learning Algorithm that combines predictions of several weak learners to create to create strong predictive model.

Gradient Boosting Regressor uses boosting technique which is a type of ensemble technique that builds model subsequently. Each model focuses on the errors made by previous models.

In context if our model the weak learners are typically decision trees with limited depth called stumps.



Advantages of Gradient Boosting Regressor:

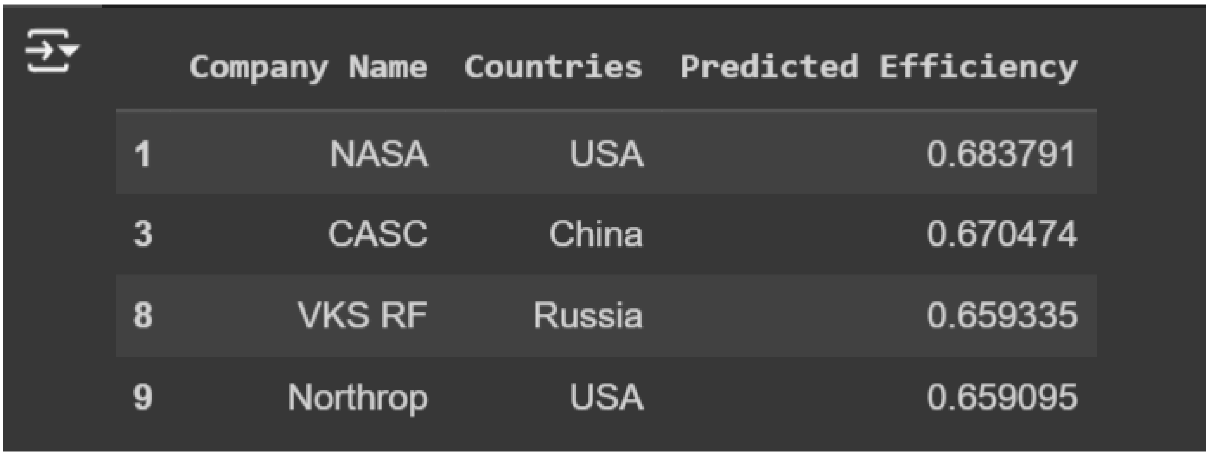
1) Produces models with higher accuracy
2) Can provide insights into importance of different features.

Using Gradient Boosting Regressor we got an accuracy of 73.57% approx, where the predicted countries and companies are:

5) **NASA - USA**
6) **CASC - China**
7) **VKS RF - Russia**
8) **Northrop - USA**

Code Snippet:

```
model_2=GradientBoostingRegressor(n_estimators=100,learning_rate=0.05,m
ax_depth=5,random_state=42)

model_2

model_2.fit(x_train,y_train)

y_pred=model_2.predict(x_test)

y_pred

model_2_final_predictions=pd.DataFrame(x_test,columns=x.columns)

model_2_final_predictions['Predicted Efficiency']=y_pred

model_2_final_predictions[['Company
Name','Countries']]=original_names.loc[x_test.index].values

model_2_final_predictions=model_2_final_predictions[['Company
Name','Countries',  'Predicted  Efficiency']].sort_values(by='Predicted
Efficiency',ascending=False)

model_2_final_predictions
```



	Company Name	Countries	Predicted Efficiency
1	NASA	USA	0.683791
3	CASC	China	0.670474
8	VKS RF	Russia	0.659335
9	Northrop	USA	0.659095

Model Evaluation Metrics:

1) Mean Squared Error:

   Average squared difference between predicted values and actual values.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

n: Number of observations
y : Actual observation
y(hat): Predicted Obervation

2) R² Error:

   Statistical measure that represents proportion of variance for a dependent variable that explained by independent variables

   Indicates how well the model fits the data.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

Where
SSres : Residual sum of squares
SStot : Total sum of squares

Code Snippet:

```
mse=mean_squared_error(y_test,y_pred)

r2=r2_score(y_test,y_pred)

accuracy=r2*100

print(f'Mean Squared Error:{mse}')

print(f'R^2 Score:{r2}')

print(f'Accuracy:{accuracy: .2f}%')
```