Executive Summary - Justin Ferrales

Project Proposal:

I have made quite a few changes since the project proposal given from phase 1. I have made the switch from pokemon go onto launch vehicle analysis. I decided to make the switch because there was such little data on Pokemon Go so I started researching other topics and launch vehicle analysis lines up with my interests as I am an aerospace major. The goals I hope to complete with this project is to see the reliability and the success of different rockets, while varying country, launch site, and production costs. Later in my work, I plan on looking into rocket reusability, e.g. SpaceX's boosters, and do some analysis to see if rocket reusability in the future can be done.

Data Sources:

I have two main sources as of right now, but I plan on scraping a SpaceX API later, or finding a Kaggle dataset on the topic. I scraped a webpage "[https://nextspaceflight.com/rockets/"](https://nextspaceflight.com/rockets/). In it I was able to get rocket name, number of missions, successes, partial failures, failures, success streak, and success rate. To do this I used beautiful soup to search for tags to get the aforementioned rocket elements above. Additionally, I used a kaggle dataset "[https://www.kaggle.com/code/isaienkov/space-missions-eda-time-series-anaysis/input"](https://www.kaggle.com/code/isaienkov/space-missions-eda-time-series-anaysis/input), which I will be using to compare costs and get data about launch site placement and time of flight. The observational unit in the first one is a singular rocket. The observational unit in the second one is a launch mission.

Main Variables of Interest

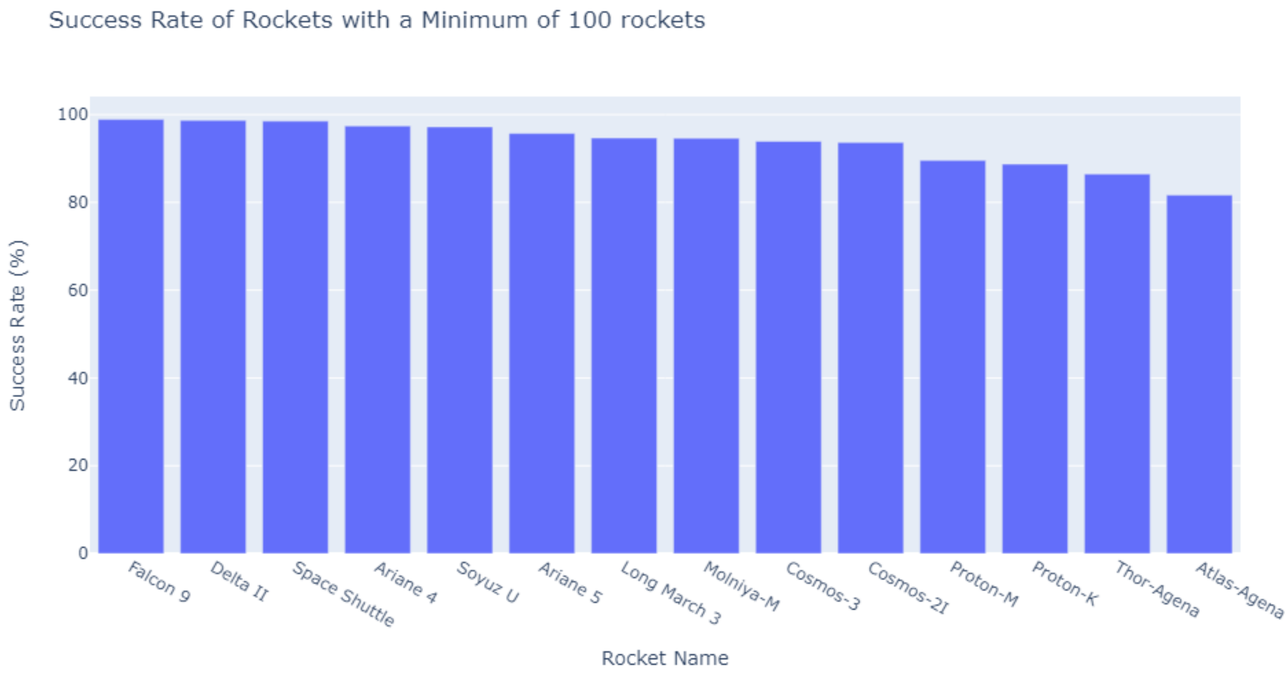
I plan on using all of the columns stated above to do my analysis. The website I scraped from has data dating back to the first rocket launch, Sputnik, all the way up to current times. They even have rocket launches planned, even one close to Cal Poly, as SpaceX will launch out of Vanderberg on Saturday 11/11.In my second dataset, there was such littler variables and I did additional research to get more values for machine learning later on.

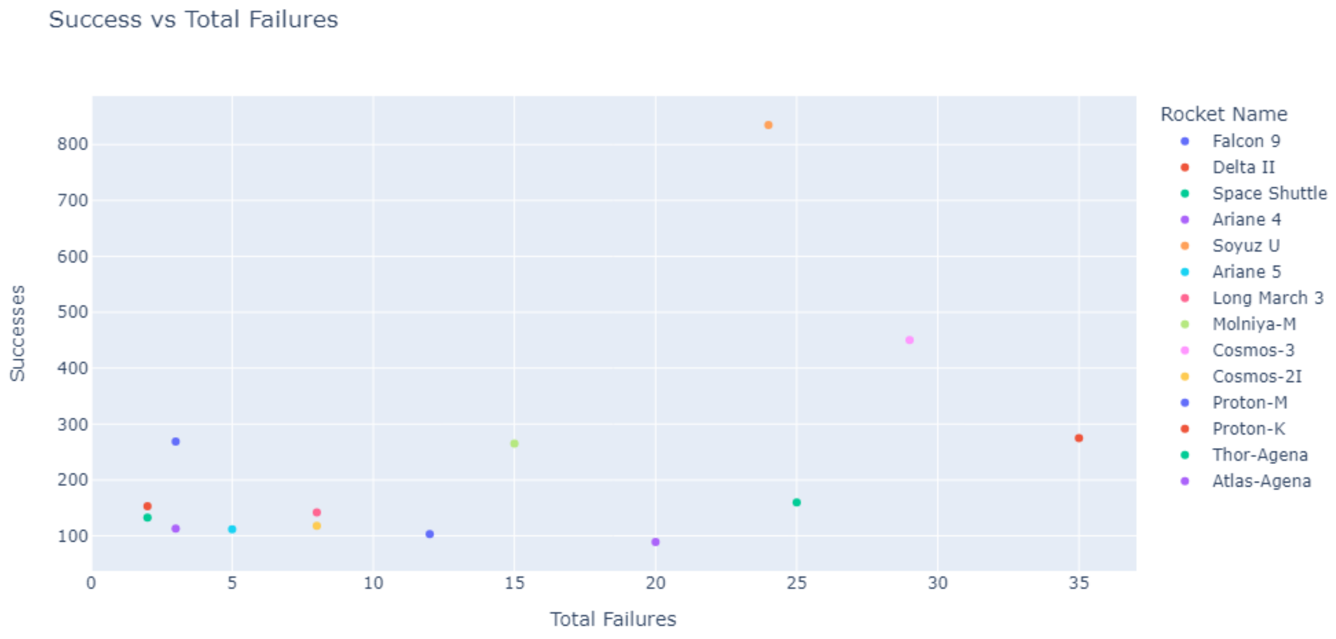
Data Cleaning/Processsing

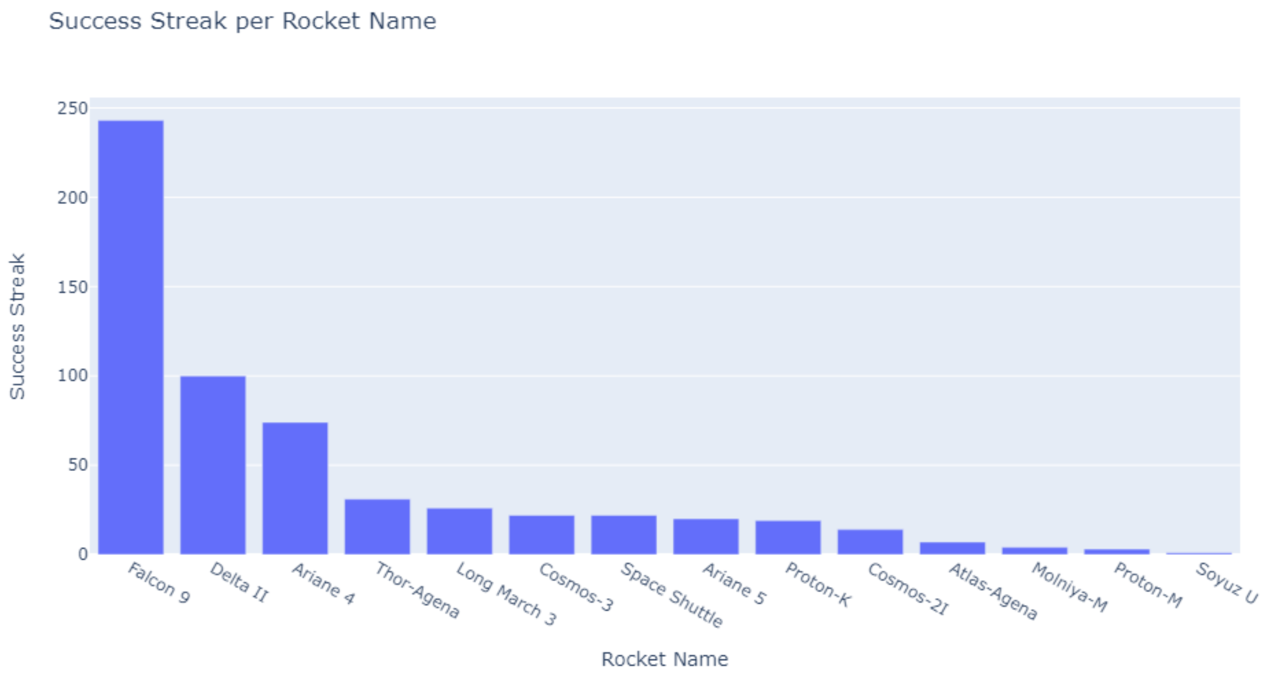
On my website scraping, I found that they used total failure and total success as a metric to compute success rate. However, I made a new column to make the total and partial failure, failure in total, and used the success to compute an accurate success rate.

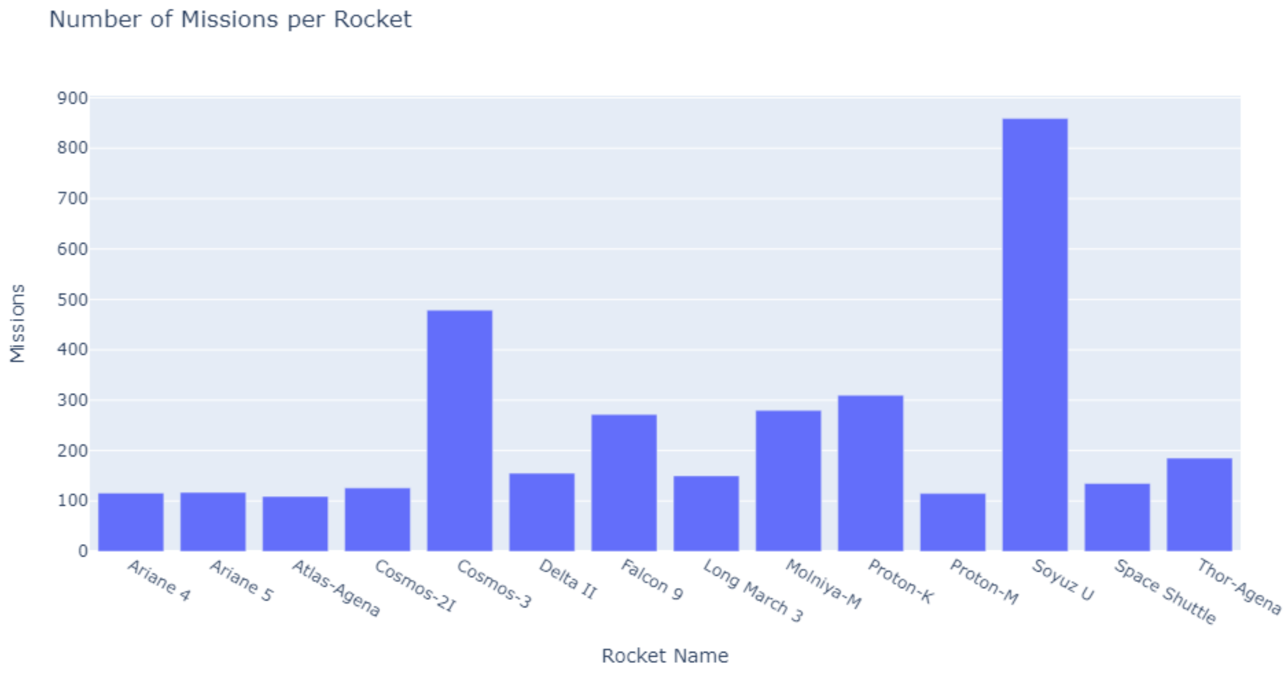
Success Rate = # of Successes / (# of Partial Failure + # of Failu

In terms of data cleaning, I filled in all of the zeros in both the dataframes I am using. I am going to go down the list of what I have done so far. From the web scraping portion, I made several data frames for the various web pages, and I use pd.concat to concatenate all of the dataframes together. For both datasets, all of the data were casted as objects, so to do quantitative analysis on the columns I type casted them into integers. I had a lot of data on rockets, so I chose ones with high TRL (Tehcnology Readiness Level) and used number of launches as a TRL metric. I only used rockets with over 100 missions, for the first dataset.




Success vs Total Failures
Successes
Total Failures
800
700
600
500
400
300
200
100
0
5
10
15
20
25
30
35
Rocket Name
Falcon 9
Delta II
Space Shuttle
Ariane 4
Soyuz U
Ariane 5
Long March 3
Molniya-M
Cosmos-3
Cosmos-2I
Proton-M
Proton-K
Thor-Agena
Atlas-Agena


Success Streak per Rocket Name
Success Streak
Rocket Name
250
200
150
100
50
0
Falcon 9
Delta II
Ariane 4
Thor-Agena
Long March 3
Cosmos-3
Space Shuttle
Ariane 5
Proton-K
Cosmos-2I
Atlas-Agena
Molniya-M
Proton-M
Soyuz U



For the next dataset, I made various new columns to compute different things down the line. I wanted to see how the US performed against the world so I did some mapping like how we mapped passengers by class in the titanic data. I did that quite a bit, and here are some of the columns I came up with.

def values(c): val = c.strip() if 'USA' in c: return "USA" else: return "World" space["US_or_Not"] = space["Location"].map(values)

def successCount(c): if "Success" in c: return 1 else: return 0

space["binary_success"] = space["Status Mission"].map(successCount)

def launchPad(c): val = c.strip() p = val.split(',') return p[0]

space["Launch Pad"] = space["Location"].map(launchPad)

def launchCenter(c): val = c.strip() p = val.split(',') return p[1]

space["Launch Center"] = space["Location"].map(launchCenter)

def country(c): val = c.strip() p = val.split(',') if p[-1] == " New Mexico": return "USA" return p[-1]

space["Country"] = space["Location"].map(country)

def stateUS(c): val = c.strip() if "USA" in c: p = val.split(",") return p[2].strip() else: pass

space["State"] = space["Location"].map(stateUS)

def vehicle(c): val = c.split("|") return val[0]

space["Vehicle"] = space["Detail"].map(vehicle)

def DayofWeek(c): val = c.split() return val[0] space["Day of the Week"] = space["Datum"].map(DayofWeek)

def month(c): val = c.split() return val[1]

space["Month"] = space["Datum"].map(month)

def day(c): val = c.split() new_string = val[2].replace(",", "") return new_string

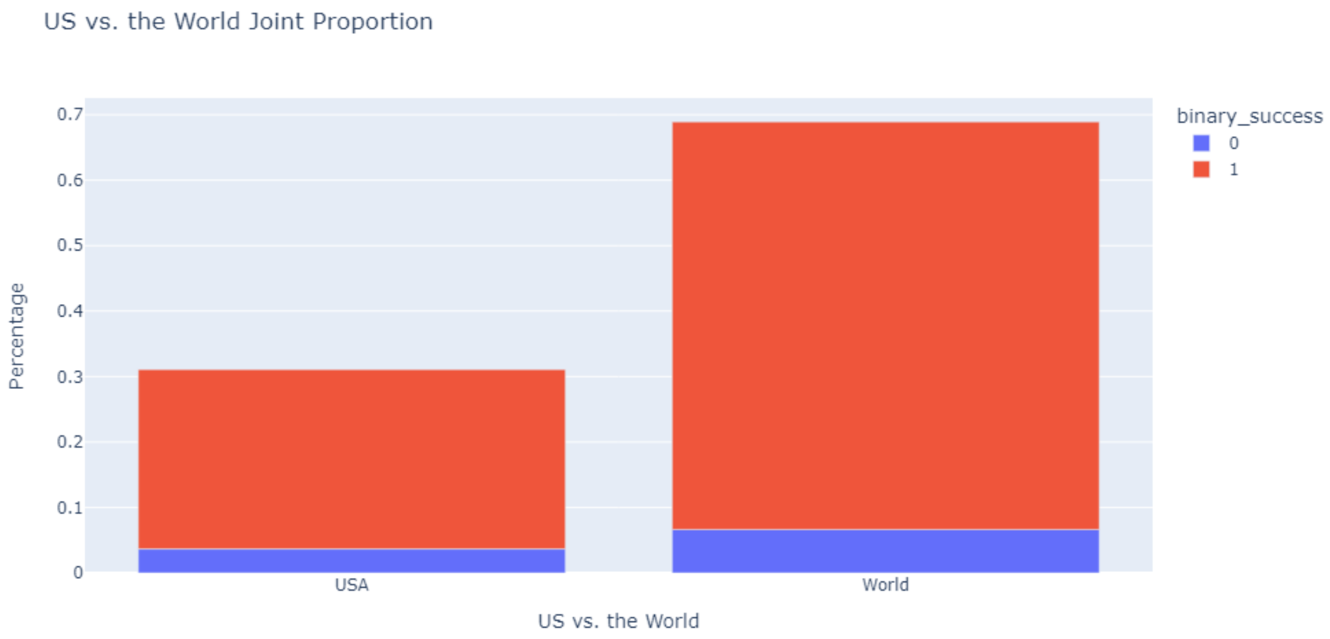
space["Day"] = space["Datum"].map(day)

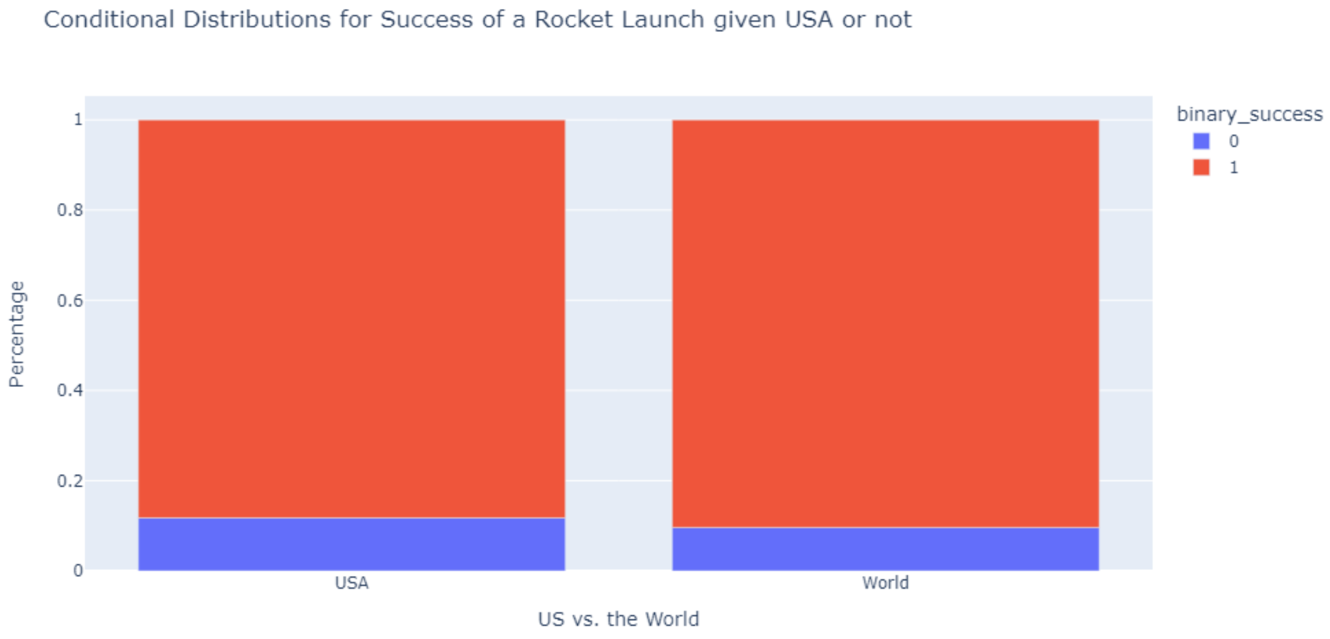
def type_agency(c): gov = ["CASIC", "Khrunichev", "CASC", "Roscosmos","JAXA", "VKS RF", "ISRO", "KARI","RVSN USSR","AMBA", "ESA", "NASA", "AEB", "US Air Force", "CNES", "RAE", "Armée de l'Air", "US Navy"] if c in gov: return "Government" else: return "Commercial"

space["Agency Type"] = space["Company Name"].map(type_agency)

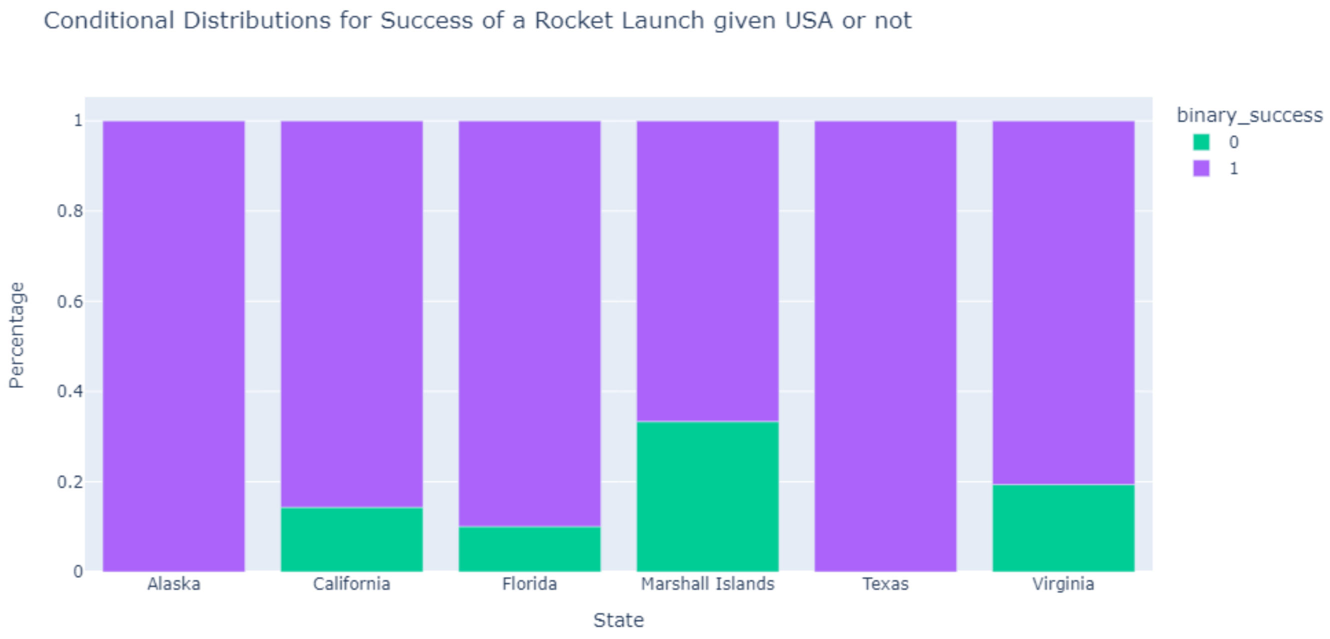
def numSatellites(c): count = 1 val = c.split("|") for item in val[1]: if item == "&": count += 1 return count space["Satellite Count"] = space["Detail"].map(numSatellites)

After that I computed some analysis using crosstab and groupby to compute the success of the US rockets. From the plots the US contributed 27.4% of total successes in the total history of rocketry. In terms of success rate the US has an 88% success rate which is 2% lower than the world's success rate at 90%.

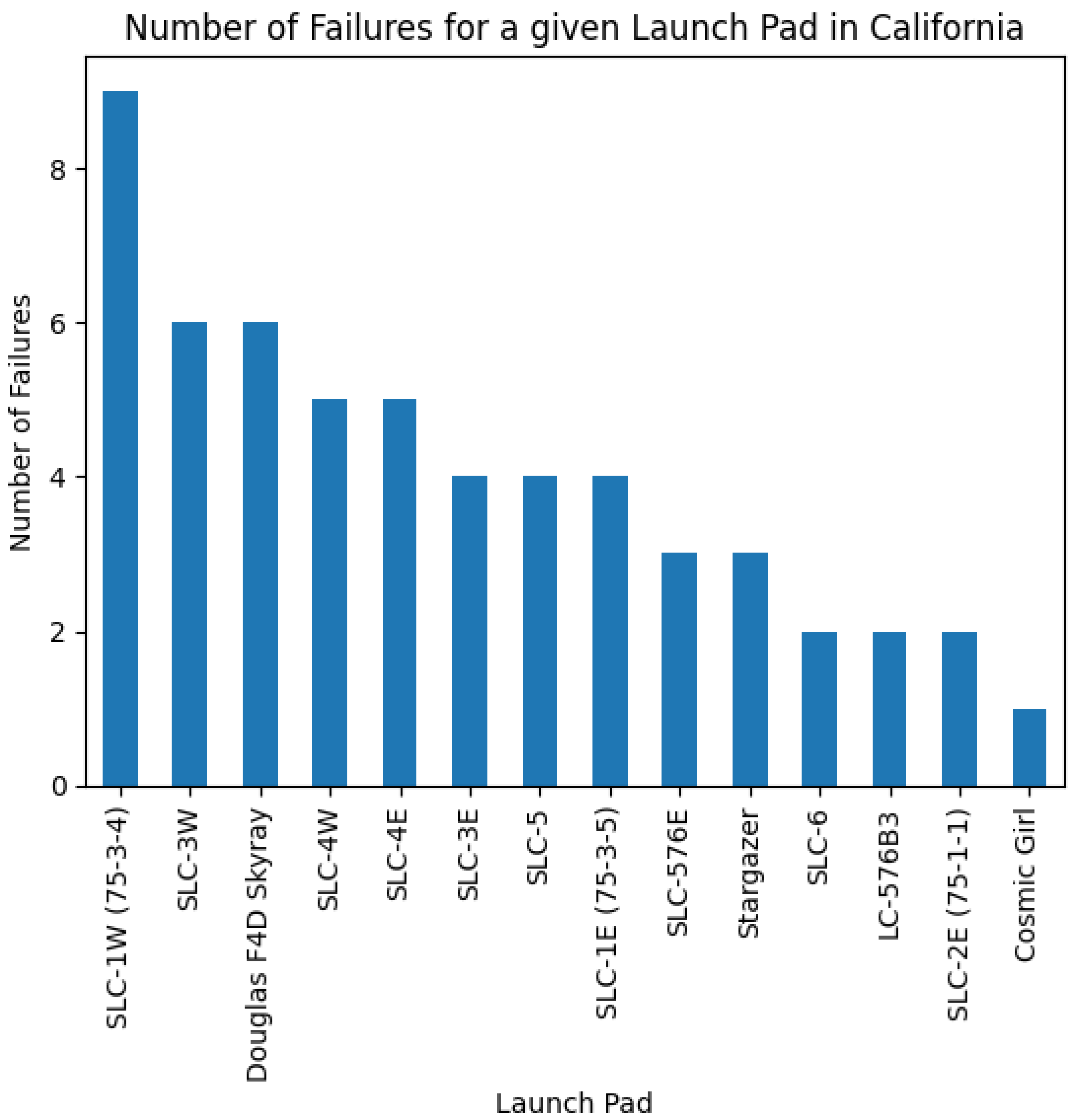




Keeping with the theme of US success, I wanted to see how the different launch sites affected reliability.



This data can be taken out of context as Alaska only had a few launches relative to the other states. CAlifornia and Florida are space hubs so their success rate is the bulk of the analysis. With that being said, I looked at the different launch pads in California, primarily here in Vanderberg Space Force Site (SLC).



Part 3 Machine Learning

Description of the model

I started off by getting my data that did not have any NaN's namely in the Cost column of my dataframe. Following this I ended up using KNeighborsClassifier and using the columns: Day, Satellite Count, Cost, Agency Type, Day of the Week, and the thing I wanted to test was the success of the mission. For the categorical values I used a one hot encoder and defined a pipeline, using KNeighborsClassifier. I also did a seperate model predicting the cost of a rocket per its satellite count, country of origin, month, and agency type, i.e. commericial or government entity.

This is for the first model described

from sklearn.preprocessing import StandardScaler from sklearn.neighbors import KNeighborsClassifier from sklearn.pipeline import make_pipeline

from sklearn.compose import make_column_transformer from sklearn.preprocessing import OneHotEncoder

x_train = df_ml[["Day", "Satellite Count", "Cost", "Agency Type", "Day of the Week"]] x_test = df_test[["Day","Agency Type", "Day of the Week"]] y_train = df_ml["Status Mission"]

ct = make_column_transformer((OneHotEncoder(), ["Agency Type", "Day of the Week"]), remainder="drop" # all other columns in X will be dropped.)

pipeline = make_pipeline(ct, KNeighborsClassifier(n_neighbors=5))

pipeline.fit(x_train, y_train)

Second Model

ct = make_column_transformer((StandardScaler(), ["Satellite Count"]), (OneHotEncoder(), ["Country", "Month", "Agency Type"]), remainder = "drop")

from sklearn.pipeline import make_pipeline from sklearn.neighbors import KNeighborsRegressor pipeline1 = make_pipeline(ct, KNeighborsRegressor(n_neighbors=5))

pipeline1.fit(X=df_ml[["Satellite Count","Country", "Month", "Agency Type"]], y=df_ml["Cost"])

Description of new models

1st model

```
import pandas as pd
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import make_pipeline
from sklearn.compose import make_column_transformer

ct = make_column_transformer(
    (OneHotEncoder(handle_unknown='ignore'), ["Month", "Country", "Agency Type"]), remainder = "passthrough"
)

# this is my training data

xTT = space[["Month", "Country", "Agency Type", "Satellite Count"]]
yTT = space["binary_success"]

# define a pipeline
pipeline20 = make_pipeline(
    ct,
    KNeighborsClassifier(n_neighbors=2)
)

pipeline20.fit(xTT, yTT)
a = pipeline20.predict(xTT)
pd.Series(a).value_counts()
```

```
1    3815
0     509
dtype: int64
```

2nd model

```
ct = make_column_transformer(
    (OneHotEncoder(), ["Month", "Day of the Week", "Agency Type"]), remainder = "passthrough"
)

# this is my training data

xT1 = space[["Month", "Day of the Week", "Agency Type", "Satellite Count"]]
yT1 = space["binary_success"]
pipeline0 = make_pipeline(
    ct,
    KNeighborsClassifier(n_neighbors=2)
)

pipeline0.fit(xT1, yT1)
```

```
Pipeline
```

3rd Model

```
ct_new = make_column_transformer(
    (OneHotEncoder(), ["Month", "Day of the Week"]), remainder = "passthrough"
)

# this is my training data

xT_new = space[["Month", "Day of the Week"]]
yT_new = space["binary_success_label"]

# define a pipeline
pipeline_new = make_pipeline(
    ct_new,
    KNeighborsClassifier(n_neighbors=2)
)

pipeline_new.fit(xT_new, yT_new)
```

I also performed grid search to find the best k to use for the model.

Double-click (or enter) to edit

Results

Overall, my results were saddening. For my first model, it always predicted a successful launch which is not the case. In the dataframe there were 400 failures, so there should be some percentage of failure. I think it is due to the variability of the reasons why a rocket failed. Trying to predict whether or not a rocket failed based on day, number of satellites, cost, agency type, and day of the week are not appropriate metrics to determine the success of a rocket