Launch Vehicle Analysis and Machine Learning

Justin Ferrales

Motivation

- I chose this project because I am an aerospace engineering major.
- Perform data analysis on various launch vehicles and their associated space missions.
- Discover trends between various variables within both dataframes.
- Utilize machine learning on varying feature to predict whether or not a mission will be successful or not

Fig 1: SpaceX Prototype Rocket

Data Sets

For this project, I used a dataset from kaggle: (https://www.kaggle.com/code/isaienkov/space-missions-eda-time-series-anaysis) and I scraped data from this website: https://nextspaceflight.com/rockets/?page=7&search=

The observational unit in the dataset from kaggle was all the individual space missions. The observational unit from nextspaceflight was about individual launch vehicles.

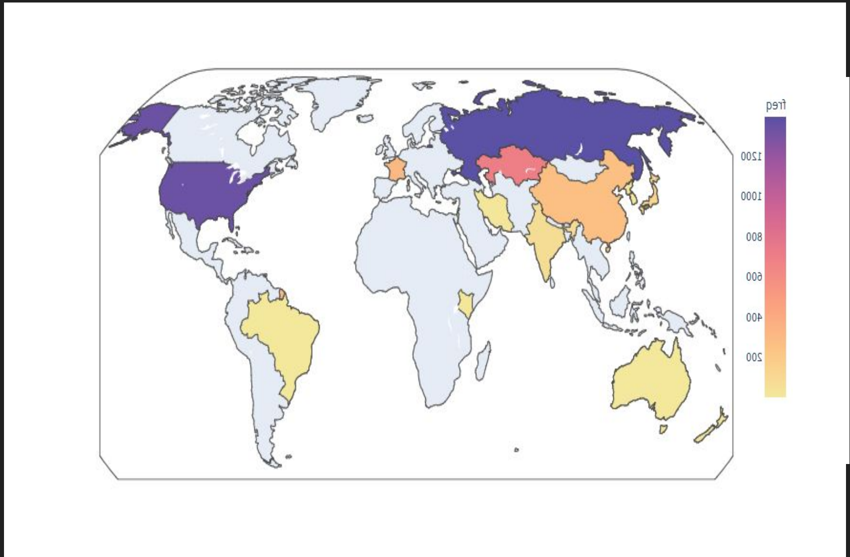


Fig 1: Choropleth map created from kaggle dataset. Number of launches per country.

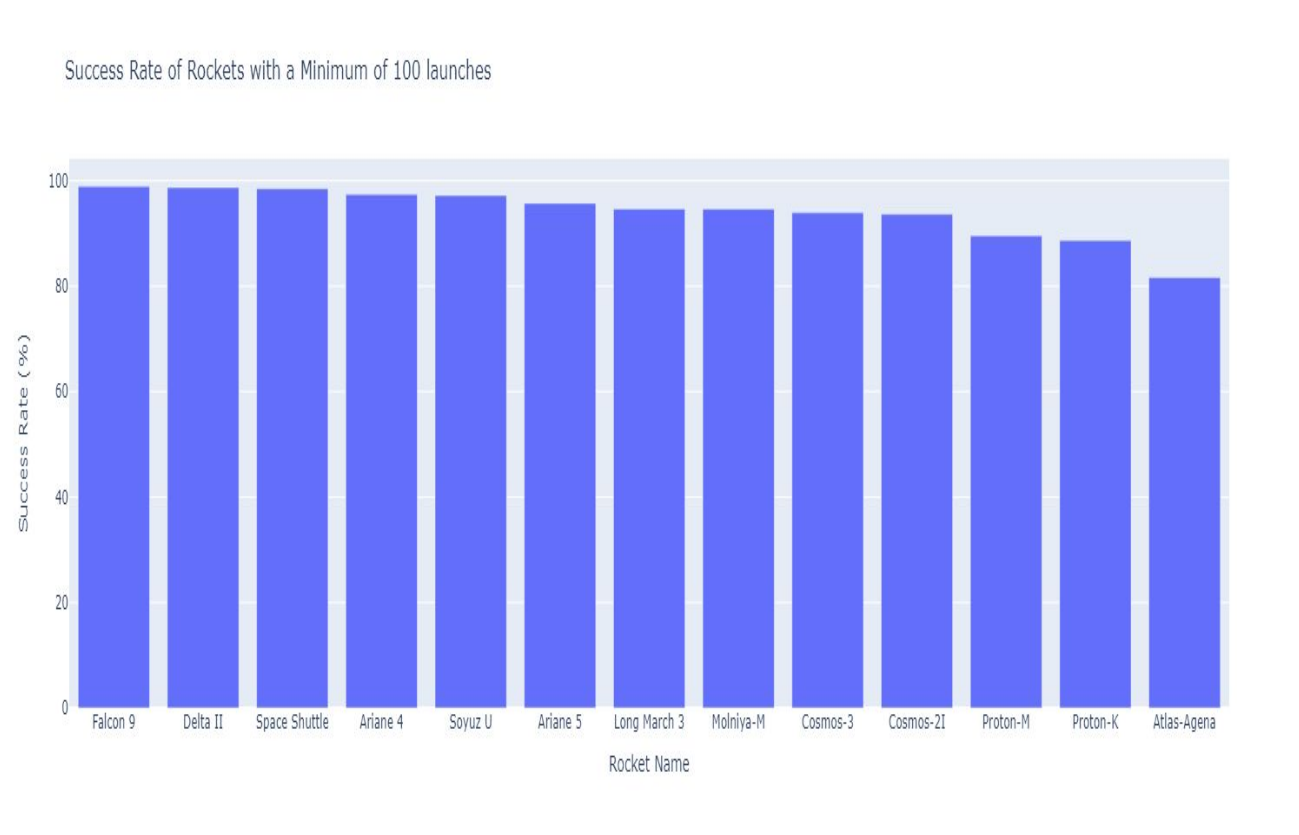


Fig 2: Success Rate of Rockets with a Minimum of 100 launches

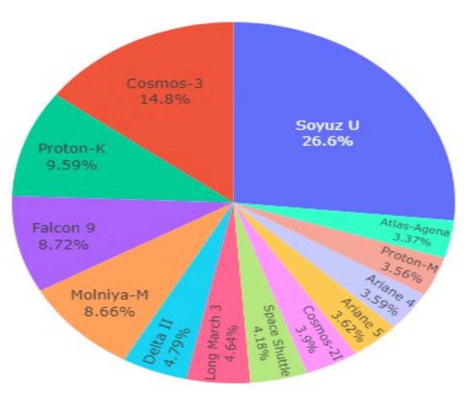
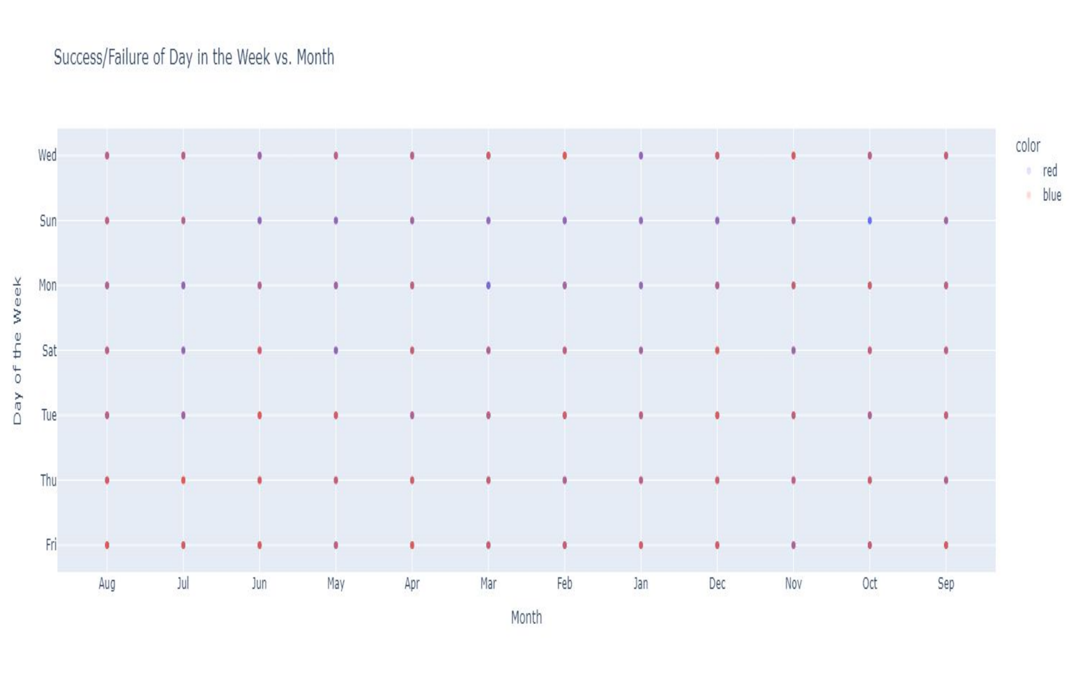
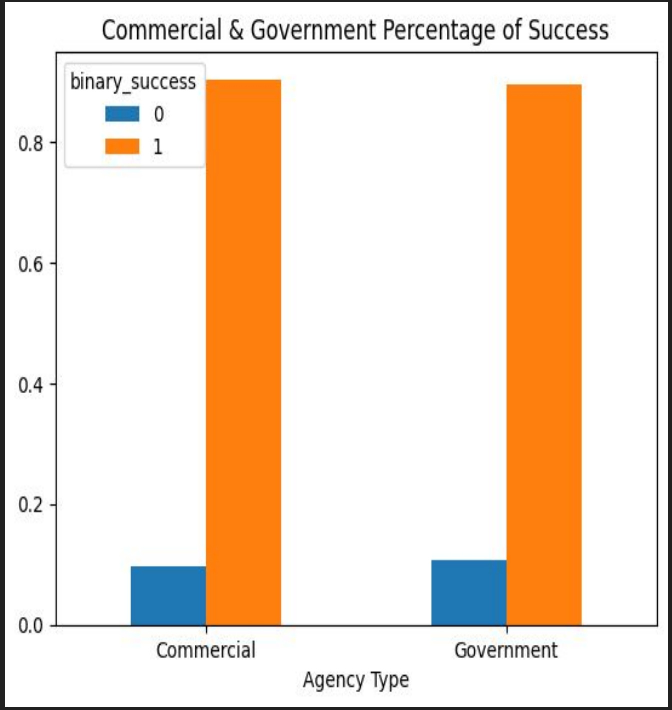


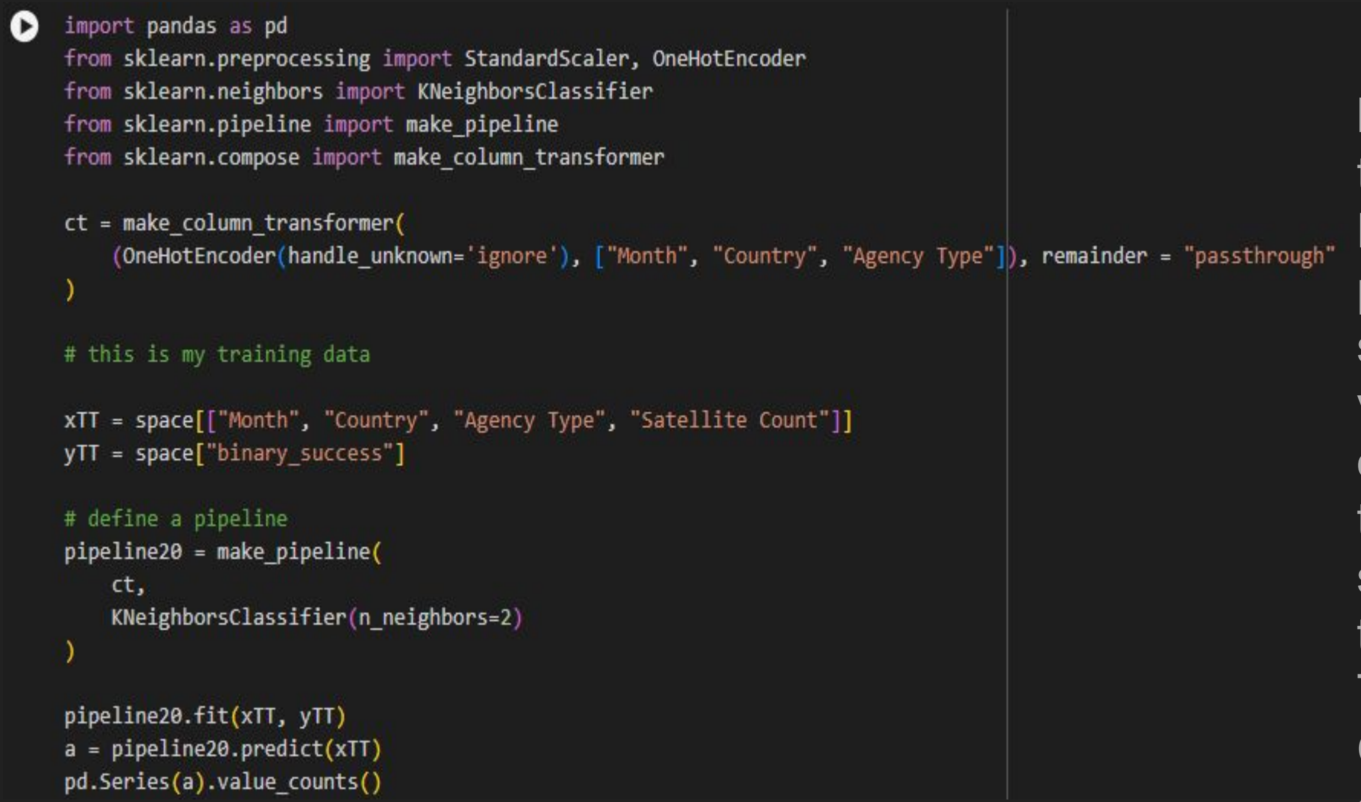
Fig 3: Proportion of launches per rocket relative to rocket with high TRL.





From my second dataset (the one I will be doing machine learning on), some key features that I thought were important are success/failure of Day in the Week vs. Month and whether or not it is a commercial or government agency..

Machine Learning 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()
```

I ran 3 machine learning models, all 3 of them using KNeighborsClassifier predicting success/failure for varying features. The other two models were trained on: space[[“Month”, “Day of the Week”, “Agency Type”, “Satellite Count”]], space[[“Month”, “Day of the Week”]]

Machine Learning Model Results

	Precision for Success	Recall for Success	F1 Score for Success	Precision for Failure	Recall for Failure	F1 Score for Failure
Model 1	0.896	0.893	0.8944	0.8	0	0
Model 2	0.893	0.846	0.868	0.453	0.006	0.0122
Model 3	1.0	1.0	1.0	1.0	1.0	1.0

Machine Learning Conclusion

With the exception of model 3, my models have a high precision and recall score for success, but have a very low precision and recall score for failure. This could be due to a multitude of reasons. It is hard to predict whether or not, a rocket with fail based on the features chosen. Additionally, the rocket dataset from kaggle had 3618 successes and 400 failures. This could have skewed our model because our model has more successes to predict from relative to failures. A rocket can fail due to malfunctions between hardware, improper design, and lack of in-depth testing, etc. All of which were not features available in the dataset from Kaggle.