Predicting the landing outcome of the first stage of a Falcon 9 using data science techniques and machine learning

By Nandini Tejas Mehta

45939400

DATA9703 Capstone Project, The University of Queensland

5th June 2024

Abstract

This project focuses on using historical data from Falcon 9 rocket launches to develop a machine learning (ML) model that is capable of predicting the landing outcomes of its first stage. The Falcon 9's reusability, achieved through its ability to land back on Earth, significantly reduces launch costs and enhances the feasibility of future space missions. By creating a predictive model, this project aims to provide SpaceX with critical insights into the factors influencing successful landings which would optimize future mission planning and reliability.

Through extensive data analysis and the application of various ML techniques, the Random Forest model emerged as the most accurate, achieving an impressive 94.87% accuracy. Key features impacting landing success were identified, including the core used, payload mass, core reuse frequency, and the type of orbit targeted. These findings offer SpaceX with valuable guidelines to enhance mission success rates by focusing on optimal core specifications, payload masses, and orbit types which will help advance cost-effective and sustainable space exploration for the future.

Executive Summary:

Project Aim and Rationale

Given Falcon 9's unique capability of landing back on Earth and being reused in future missions, accurate prediction tools would be useful to have as they could ensure the success of landing which would help plan space missions. This initiative seeks to create a data-driven solution that provides SpaceX with valuable insights to optimize mission planning, improving reliability and efficiency to increase the reliability and make future space missions more cost efficient.

Methodology

The project involved a comprehensive analysis of Falcon 9's historical data using various data science techniques. Key steps included:

1. **Exploratory Data Analysis (EDA):** To understand the factors influencing landing outcomes.
2. **Interactive Folium Map:** To support EDA and understanding of launch sites
3. **Machine Learning Model Development**: Evaluating multiple models including Random Forest, XGBoost, and K-Nearest Neighbors (KNN).
4. **Feature Importance Analysis:** Identifying critical variables affecting landing success.

Results and Findings

Among the tested models, the Random Forest model exhibited the highest accuracy at 94.87%, followed by XGBoost at 89.74%, and KNN at 61.53%. The Random Forest model not only minimized overall errors but also had zero false positives, making it the most reliable for predicting successful landings. Key features influencing landing outcomes included:

- Core Used: Certain cores, such as core: 33, had higher success rates.
- Payload Mass: Optimal success was associated with payload masses weighing around 10,000 kg.
- Core Reuse Frequency: Cores reused 1 time showed significant success
- Orbit Type: Orbits such as GTO, ISS, and LEO had higher success rates, indicating potential preferences for specific mission types.

Conclusions and Implications

The insights derived from this analysis provide SpaceX with actionable guidelines for enhancing Falcon 9’s mission planning. By focusing on specific core specifications, optimal payload masses, and preferred orbit types, SpaceX can improve the reliability and cost-efficiency of its launches.

Table of Contents:

1. Introduction:

The aim of this project to use Falcon 9's historical data and create a machine learning (ML) model that can predict the outcome of the first-stage. This project will focus on using data science techniques to analyse the data and create the predictive model. Next, I will briefly outline the Falcon 9's importance and the rationale behind this project.

The Falcon 9 is an orbital launch vehicle that has two-stages. The first stage has a mechanism that allows for it to land back on Earth. If the landing is successful the rocket core from this first-stage can be reused in future missions. This reusability aspect is the redeeming feature of a Falcon 9 as it will help reduce launch costs and make future space missions cheaper (Jones, 2018).

Thus, it is crucial to develop tools that can provide accurately predict the landing outcomes of a Falcon 9. Such tools will also provide SpaceX with more information on the characteristics of the Falcon 9 that contribute greatly towards the successful outcomes. The overarching purpose of this project is to create a data solution that helps scientists take decisions related to Falcon 9 missions with increased precision and reliability. This report contains a detailed description of how this purpose is going to be achieved. The contents of the report will be outlined next.

Firstly, I will provide the background information needed to understand the basis of this project. The background information section will contain mentions of previously-produced, related works that are based on creating prediction models for space missions. It will also outline the main objectives of this project. Secondly, the methodology of the project will be discussed in detail. This section will outline the how each project objective was fulfilled, including the data science and machine learning techniques used. Thirdly, I will provide the results and findings of this project. Lastly, I will summarize and conclude the report. The next section will expand on the background information related to the Falcon 9 and the use of ML models in space exploration.

2. Background information:

This section provides a deeper understanding of the background information for this project. Initially, it will cover the various elements of a Falcon 9. Following that, it will delve into how ML models are employed within the industry to forecast mission outcomes.

As mentioned previously, the Falcon 9 contains two-stages. The second stage contains the rocket payload which is transported into its desired orbit (Zhou & Pham, 2023). Here, the payload refers to the cargo that Falcon is designed launch into orbit. The first stage is made up of the rocket core and fuel compartment which are designed to vertically land back on Earth (Sippel et al., 2017).

This allows SpaceX to reuse the rockets core and makes future missions more economical. Thus, it is important first stage lands back on Earth successfully. The next subsection lists some literature that has been written on rockets that are designed to land and be reused.

2.1. Related Works:

The Predictor-corrector entry guidance techniques are a method for directing a rocket to land vertically (Bojun et al., 2019). However, these techniques cannot determine if the landing outcome will be successful. SpaceX currently does not use any publicly disclosed software or techniques that are able to predict the landing outcomes of the Falcon 9. This led to the assumption that a custom prediction model for the Falcon 9 is yet to be developed.

Research by Lu (2008) explains that predictor-corrector methods only guide and adjust a rocket's trajectory during landing to ensure a successful outcome. The predictive entry guidance method does not account for rocket component specifications and characteristics to predict landing success. Therefore, developing a predictive ML model using historical data from Falcon 9's launches will enhance the reliability of successfully executing future Falcon 9 missions. This will be achieved by identifying the correlations between various variables that affect the landing of a Falcon 9. The next section will provide a detailed explanation of the proposed solution.

3. Proposed solution:

This section outlines the proposed solution and discusses the project deliverables. The project utilizes a data-based strategy, using findings from an initial analysis to create a robust predictive model. The data on Falcon 9's past launches was sourced by scraping the SpaceX API. This data was then cleaned, analysed, and visualized to help understand the trends that different variables showed in correlation to the landing outcome. Additionally, an interactive map was created to understand the locations of the launches and their surroundings. The ML model was then created and evaluated to help us understand the variables that influence landing success the most. The following is a list of this project's deliverables:

1) Collection of SpaceX data via API scraping
2) Exploratory Data Analysis (EDA) on the cleaned dataset
3) Interactive map of the launch sites
4) Development of a robust ML model

4. Methodology:

This section is going to outline the methods that were used to achieve the project's deliverables. Firstly, the data collection and cleaning process for the EDA is going to be discussed, followed by a detailed explanation of the ML aspect of the project.

4.1. Data collected for EDA:

The historical was scraped from the SpaceX API (*SpaceX-API/Docs at Master · R-Spacex/SpaceX-API*, 2020). Postman, an API development tool, was used to GET (a function that retrieves data) the data from the API. I three JSON files related to the Falcon 9 past launches were then downloaded and loaded onto a Jupyter notebook as pandas data frames, which a versatile data structure.

Two main data frames were created: for launch data and core data. The launch data frame contained 101 rows and 31 columns and the following columns:

```
Column headings: ['flight_number', 'mission_name', 'mission_id', 'launch_year', 'launch_date_unix', 'launch_date_utc', 'launch_
date_local', 'is_tentative', 'tentative_max_precision', 'tbd', 'launch_window', 'rocket', 'ships', 'telemetry', 'launch_site',
'launch_success', 'links', 'details', 'upcoming', 'static_fire_date_utc', 'static_fire_date_unix', 'timeline', 'crew', 'launch_
failure_details', 'last_date_update', 'last_ll_launch_date', 'last_ll_update', 'last_wiki_launch_date', 'last_wiki_revision',
'last_wiki_update', 'launch_date_source']
```

Fig 1: Column headings for the launch_info data frame

The core data frame contained 69 rows and 13 columns:

```
Column headings: ['core_serial', 'block', 'status', 'original_launch', 'original_launch_unix', 'missions', 'reuse_count', 'rtls
_attempts', 'rtls_landings', 'asds_attempts', 'asds_landings', 'water_landing', 'details']
```

Fig 2: Column headings for the core_info data frame

4.2. Preparing and cleaning for EDA:

The data was then cleaned by handling the missing values. Missing values cannot be handled by ML models, so it is crucial to impute values precisely, without adding extra bias or noise to the data. The following plots shows the missing values present in the two data frames in yellow:

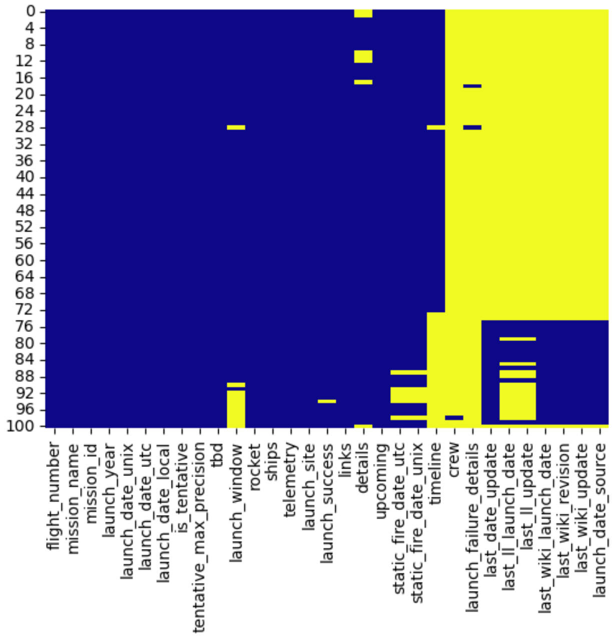


Fig 3: Missing data in launch_info

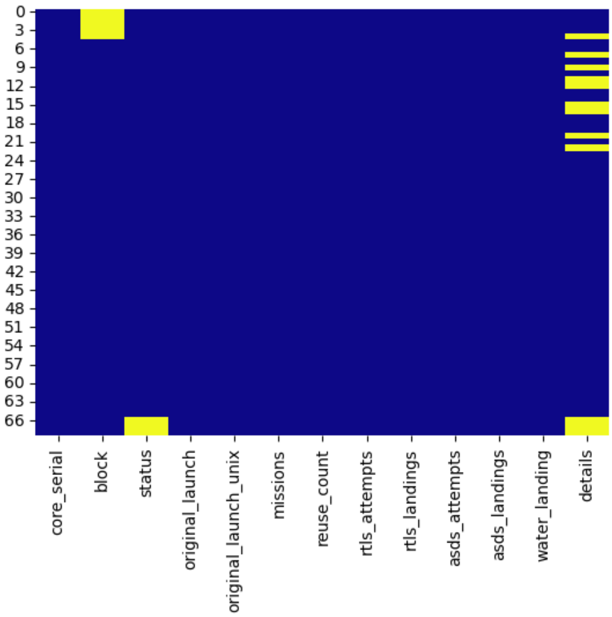


Fig 4: Missing data in core_info

Based on the plots above we can see that launch_info and core_info data frames both have missing values that needed to be handled. The important columns from Fig 1, that were used during the EDA include the ‘flight_number’, ‘launch_year’, ‘rocket’ and ‘launch_site’. The important columns from Fig 2, that were needed in the EDA includes “core_serial” and ‘resuse_count’. Therefore, all the missing data shown in Fig 3 and Fig 4 are in columns that are not essential to the EDA or the creation of the ML model, we can just delete these unwanted columns. After the missing values were handled, the data was prepped for the EDA. Firstly, the ‘rocket’ column from the launch data frame was inspected and all the important values were extracted to create new columns. Similarly, information from the 'launch_site' was also extracted. The major variables that were visualized during the EDA include:

- The impact of using gridfins on the landing outcome:

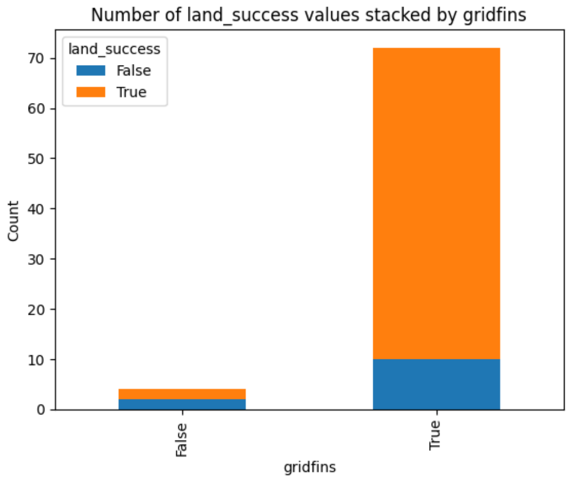


Fig 5: Number of landing success and landing failures based on the presence and absence of gridfins

- The impact of using legs on the landing outcome:

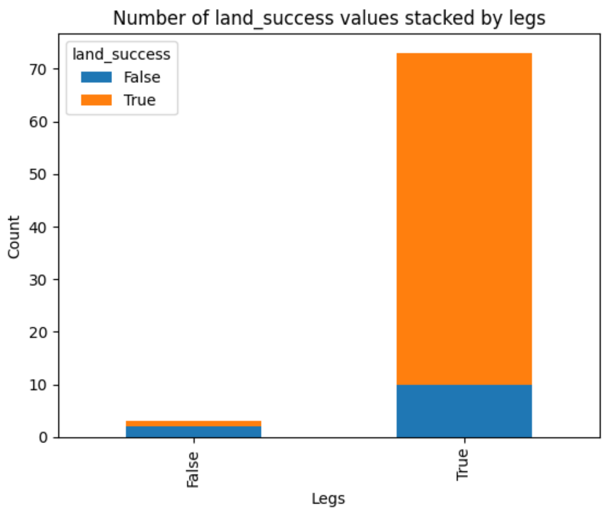


Fig 6: Number of landing success and landing failures based on the presence and absence of legs

Based on Fig 5 and 6, we can see that having gridfins and legs indicates higher levels of landing success.

- The number of times a core has been reused and its effect on the landing outcome: We can observe that lower reuse counts have a better landing success possibility. This would make sense as cores that have been reused more would be more worn out.

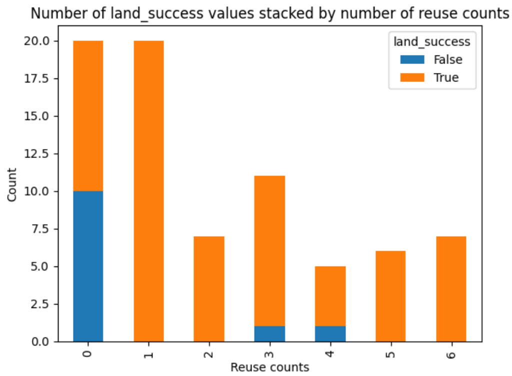


Fig 7: Number of landing success and landing failures based on the number of times a core has been reused

- The effect that landing type has on landing outcome: We can see that the ASDS (Autonomous Spaceport Drone Ship) has the highest success rates which means that this landing type could be deduced as ideal for Falcon 9 space missions.

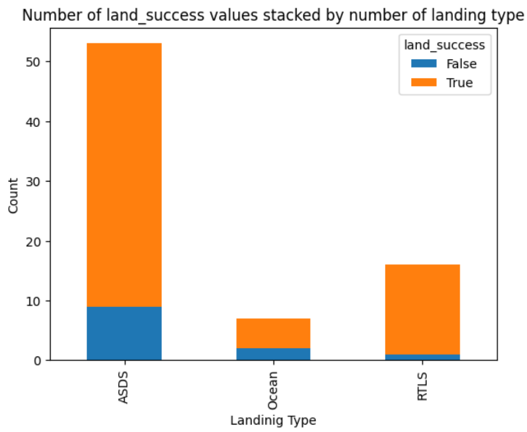


Fig 8: Number of landing success and landing failures based on the type of landing

- The effect that the landing vehicle has on the landing outcome: We can see that the OCISLY landing vehicle has the most success, thus, characteristics of this landing vehicle can be studied and applied to other landing vehicles to improve their successes.

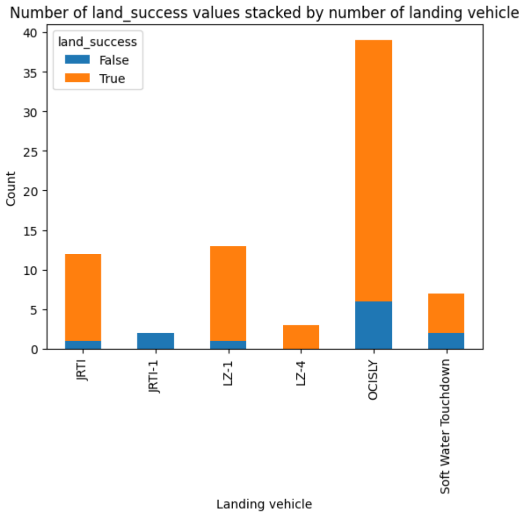
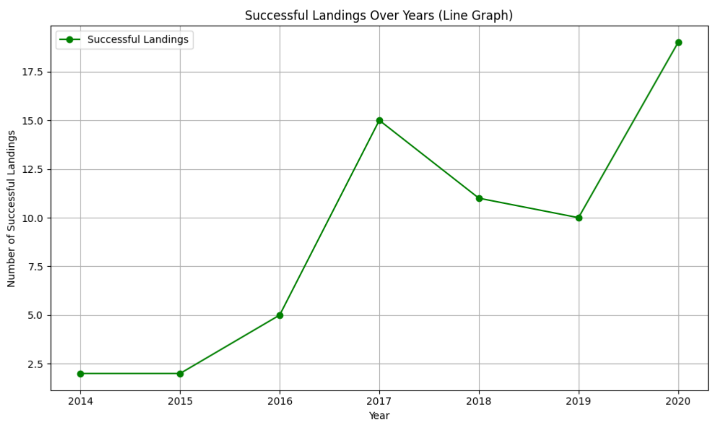


Fig 9: Number of landing success and landing failures based on the landing vehicle used

- A comparison of land success and land failures over the years: We can observe an increasing positive success rate trend. Which means that the developments on the Falcon 9 being made over the years are a step in the right direction leading to higher success rates and lower failures.



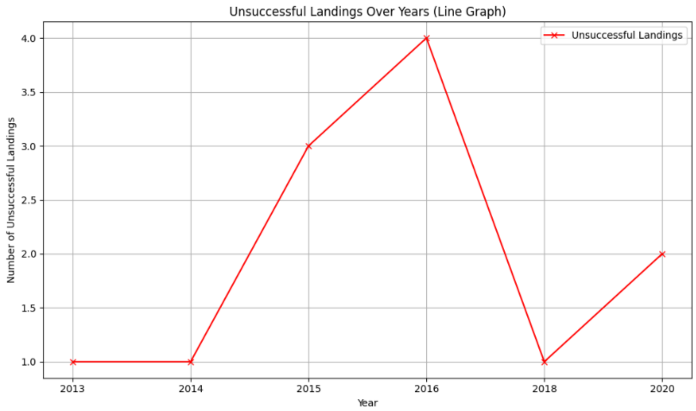


Fig 10: Number of landing success and landing failures over the years

- The effect of different orbit types on the landing outcome: The GTO orbit has the highest number of successes, which means it could be a frequently accessed orbit and that it's characteristics are ideal for the Falcon 9.

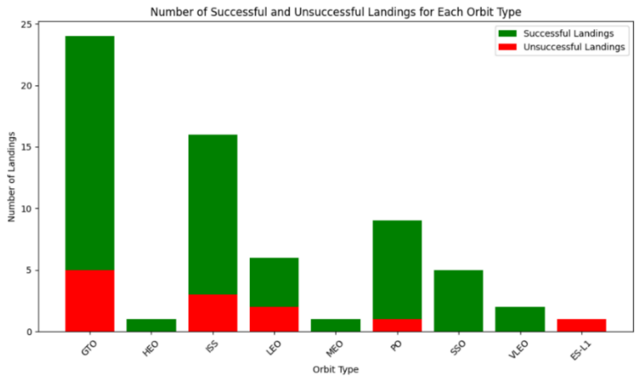


Fig 11: Number of landing success and landing failures based on the type of orbit

- The effect of payload mass on the landing outcome: Masses around 10,000 kgs have the most success which means this could be the ideal mass of payload that the Falcon 9 can carry.

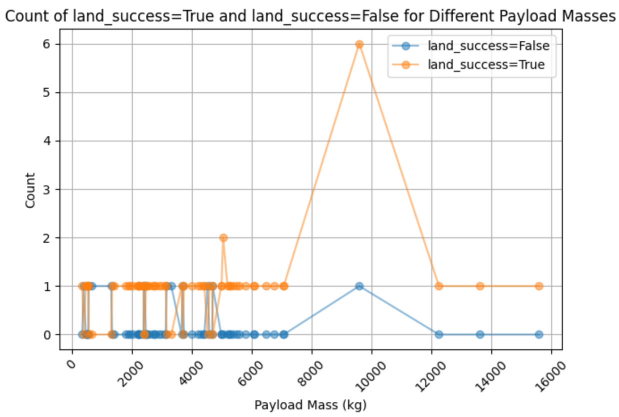


Fig 12: Number of landing success and landing failures based on the mass of the payload

- The effect of different payload types on the landing outcome:

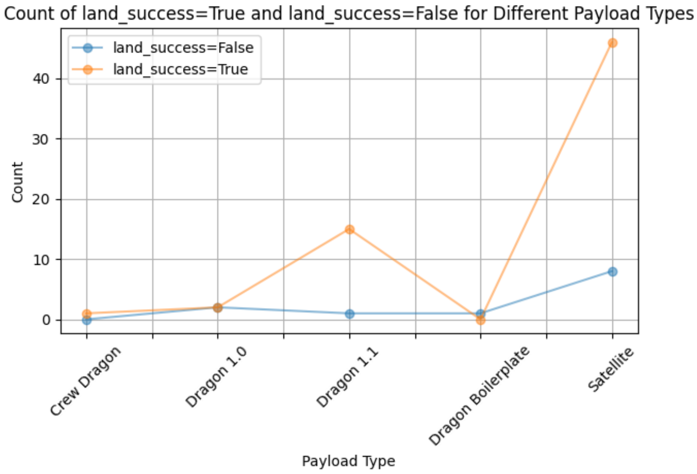


Fig 13: Number of landing success and landing failures based on the type of payload

- The effect of different launch site on the landing outcome: The Cape Canaveral Site has the highest success, its surroundings and characteristics can be studied to understand why.

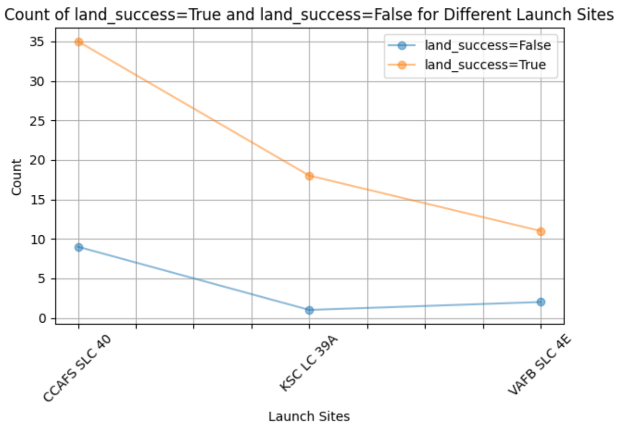


Fig 14: Number of landing success and landing failures based on the launch site

To further understand the impact of the launch sites, an interactive map was created using the Folium Python library. The coordinates for the launch sites were found using google maps. The next section briefly talks about the interactive dashboard. The following images show different aspects of the map:

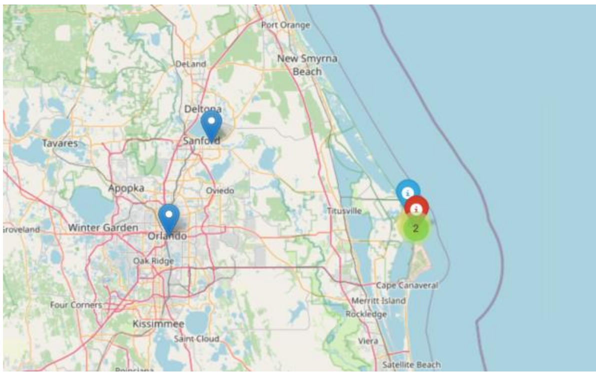


Fig 15: Shows the CCFA and KSC launch sites with the closest commercial airport in Sanford and the closest major city: Orlando.

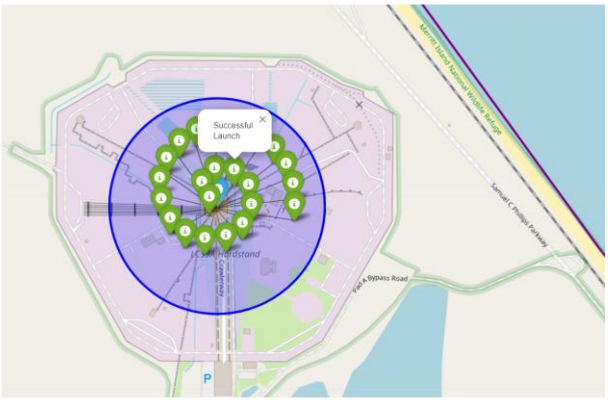


Fig 16: Shows all the successful launches at the KSC launch site along with the launch site perimeter.

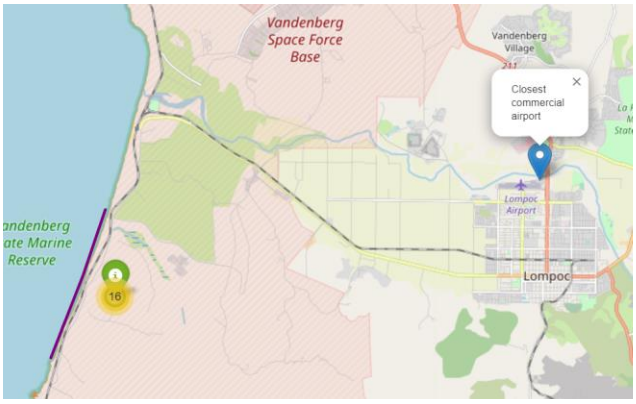


Fig 17: Shows the VAFB launch site with its number of successful launches, with the closet commercial airport in Lompac and the closest coastline marked in purple.

When the EDA visualizations are carefully analysed, we can see that all the trends in the Figures above have a lower representation for the failed landing class. This can cause problems for the machine learning aspect of this project. The next section will explain why class representation matters and how the data was prepped for the ML model.

4.3. Preparing the data for machine learning:

The aim of the machine learning model is to predict if a landing will be successful or not, depending on the characteristics of the rocket and mission. Thus, it was important to only include data from missions which had the intent to land, as if the landing intent was False, the landing outcome would also be False regardless of the mission and rocket characteristics. This would skew the ML model's predictions. The landing intent was given in the 'rockets' dictionary and was extracted to create a new column. Then, all rows with landing intent= False were deleted. Upon deleting these rows, the size of our dataset reduced to only 76 rows out of which only 12 rows belonged to the landing outcome= False.

This highlights two problems that would hinder the robustness of our ML model: a small and imbalanced dataset. If a ML model is trained using a small dataset, it would not be able to generalize its predictions well for unseen data (Reilly, 2024). Similarly, if the ML model is trained using an imbalanced dataset it would lead to a bias model that prefers the majority class over the minority class (Maheshwari S. & R.S., 2018). Thus, to overcome this problem I decided to implement a Basic SMOTE technique. The next section will expand on how SMOTE works and how its implementation affected the dataset.

4.4. SMOTE techniques and implementation on dataset:

Simple minority oversampling techniques (SMOTE) can be used to create synthetic data for the minority class, which would in turn help overcome the imbalance within the dataset and it would also increase the amount of data within our dataset which would resolve the issues related to using small datasets (Elreedy D & F., 2023). The Basic SMOTE algorithm used in this project works in the following way:

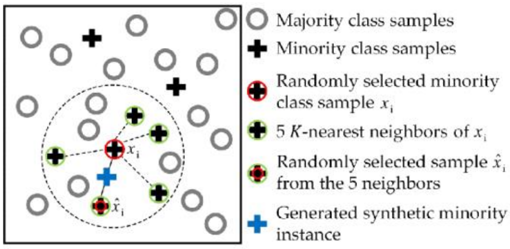
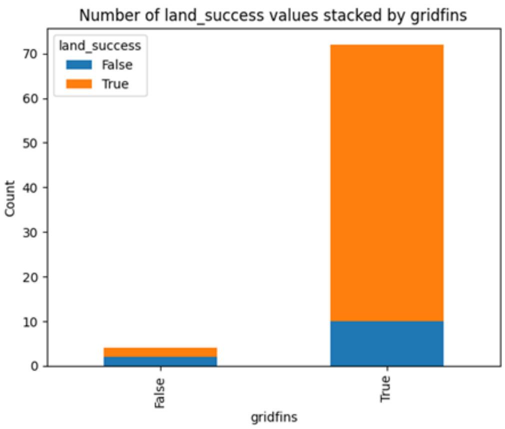
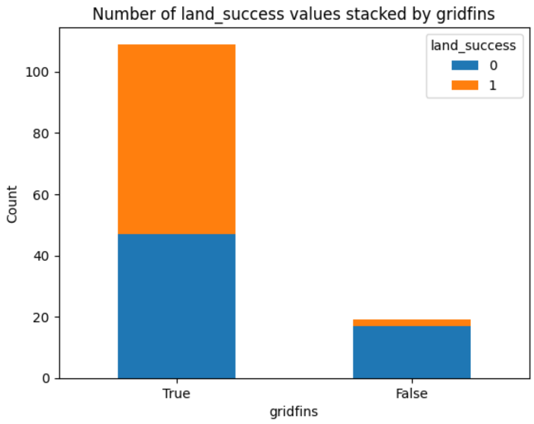


Fig 18: Shows how the Basic SMOTE algorithm works (Rich Data Admin, n.d.)

Using Fig 18 as a reference to explain the algorithm, firstly, a random data point from the minority class is selected. Next, the k-nearest neighbours of that point are found by using the Euclidian distance between them as a metric. Then a line segment is drawn between the selected minority class data point and its randomly selected neighbour. A random point on this line segment is then chosen as the synthetically generated data value. This process is repeated until the number of data samples belonging to the minority class are the same as that of the majority class. In our case, this led to the algorithm creating 52 new synthetic samples, which now meant that the number of data points belonging to the landing outcome=False class were 64 and the total number of rows in the dataset are now 128. The following compare the before and after effects of SMOTE on some of the plots from the EDA:

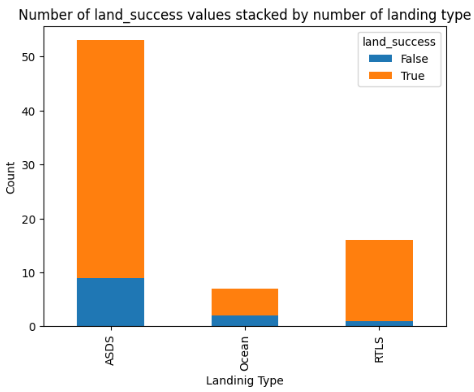


(a)

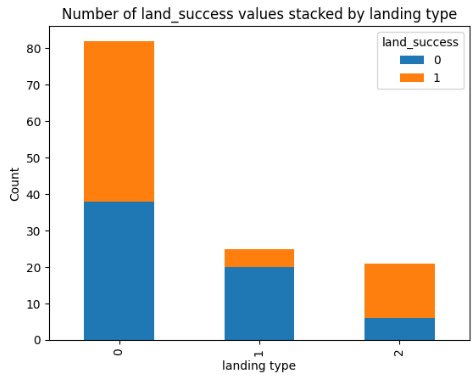


(b)

Fig 19: Plot (a) before the SMOTE, (b) after SMOTE for the impact of the presence gridfins



(a)



(b)

Fig 20: Plot (a) before the SMOTE, (b) after SMOTE for the impact of reuse count

It can be observed from both Fig 19 and 20 that plot (b) shows a more balanced distribution of both the classes (landing outcome=True or False). Thus, our data is no ready to be used in our construction of the ML model. The next subsection will elaborate on the methods that were used to obtain the best ML model for our dataset.

4.5. Machine learning model creation and evaluation:

Using the now balanced dataset, the machine learning models were now ready to be created. The idea was to implement three different machine learning algorithms: Random Forest, K-Nearest Neighbour and XGB Gradient Boost, to compare and contrast which model performed the best on the dataset. The same train-test split of 70-30 was used for all the ML algorithms, along with the same cross-validation (10-fold). This would ensure that all the algorithms have the same baseline and that this project's methodology is not biased towards any one ML method. A GridSearch was also performed to find the optimal hyperparameters for the Random Forest and KNN models. The model with the optimal parameters was then evaluated using the accuracy score metric and by analysing the model's resultant confusion matrix. The next section will go through the results obtained from each model and provide reason for selecting the optimal model. It will also provide the insights into the features from the EDA that impact the first-stage's land outcome the most.

5. Results and discussion:

Firstly, I will talk about which ML model performed the best and how this was evaluated. After which, I will discuss the deductions that we can make from the EDA to better our understanding of which variables heavily impact the landing of a Falcon 9's first-stage. The following figures show the confusion matrices obtained for each of the three models:

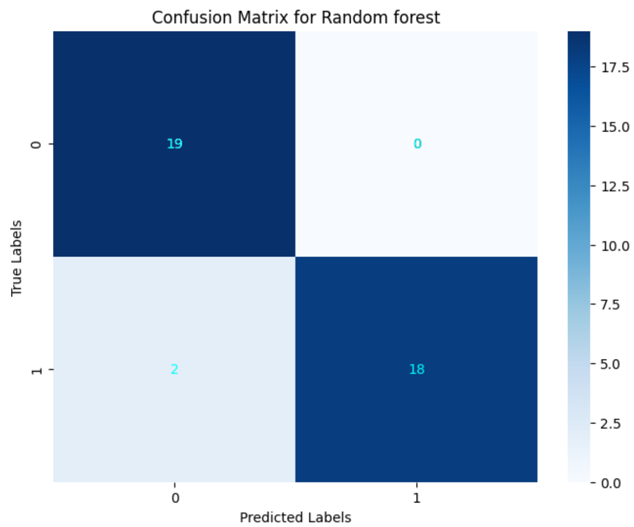


Fig 21: Confusion matrix for the Random Forest model: (TP): 18, (FP):0, (TN):19, (FN):2

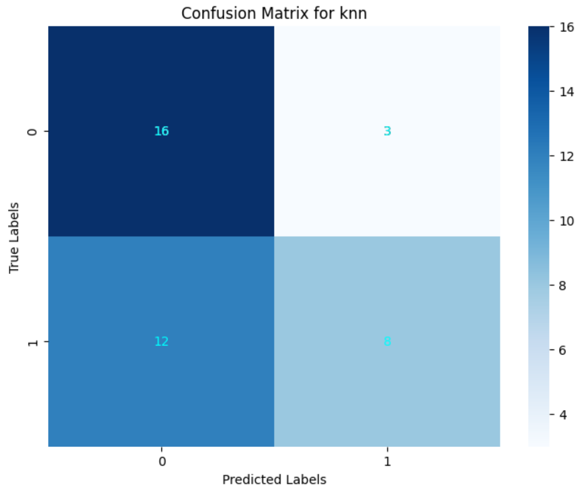


Fig 22: Confusion matrix for the KNN model: (TP):8, (FP):3, (TN):16, (FN):12

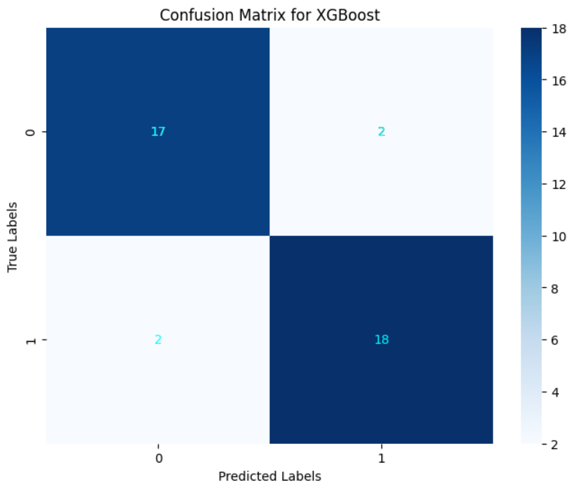


Fig 23: Confusion matrix for the XGBoost model: (TP):18, (FP):2, (TN):17, (FN):2

Accuracy scores wise, the Random Forest model performed the best with an accuracy of 94.87%, it was closely followed by the XGBoost model which had an accuracy of 89.74%. The KNN model performed the worst with an accuracy of only 61.53%.

A similar pattern can be observed from the confusion matrices shown in Fig 21, 22 and 23. The confusion matrix for the KNN model has the greatest number of misclassifications with 12 False Negatives (FN) and 3 False Positives (FP). The XGBoost model has a total of 4 misclassifications whereas, the Random Forest has the lowest number of misclassifications with only 2 FNs. Considering the fact that in our case, a FP is worse than a FN as predicting a landing to be successful when the outcome would a failure is more damaging, the Random Forest model is the best performing ML classifier as it has 0 FPs.

Based on the Random Forest model, I wanted to extract the features from the EDA that would most affect the landing outcomes of the Falcon 9. I did this using the feature_importance extractor for ML models, in Python. This gave me the following output:

```
Feature Importances:
              Feature  Importance
5         core_serial    0.216627
2     payload_mass_kg    0.152619
1         reuse_count    0.124570
8               orbit    0.100794
3         launch_year    0.098141
9        payload_type    0.068327
0              flight    0.067720
7     landing_vehicle    0.049187
4          site_names    0.040426
6        landing_type    0.030714
12               legs    0.024591
10           gridfins    0.011701
11             reused    0.010354
13     landing_intent    0.004229
```

Fig 24: Shows the features that the random forest mode has deemed important

Fig 24 shows that the most important features to consider during the planning of Falcon 9 missions, to improve the possibility of a successful first-stage landing, include the core that the first stage uses, the mass of the payload that the second-stage transports, the number of times the first-stage core has been reused, and the type of orbit the payload is to be launched into. The following plots show the trends these features exhibit after the addition of the SMOTE data:

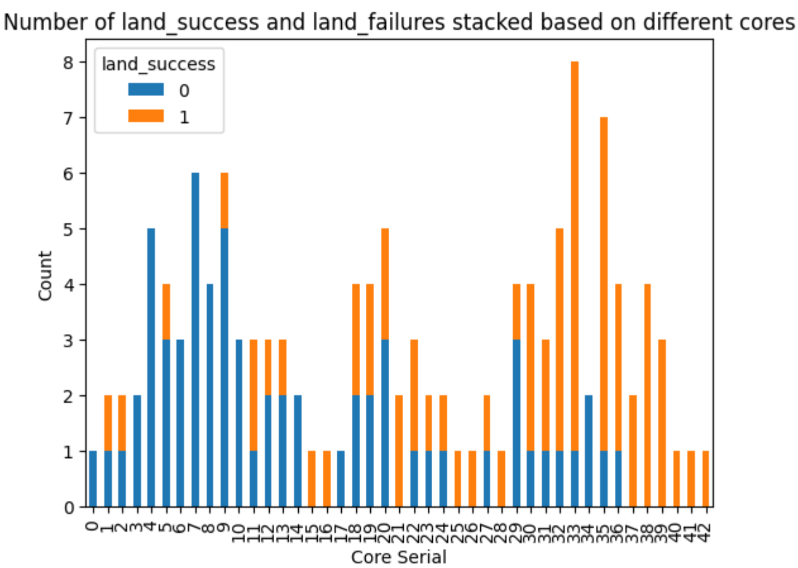


Fig 25: Showing the landing outcomes based on the core used by the first-stage

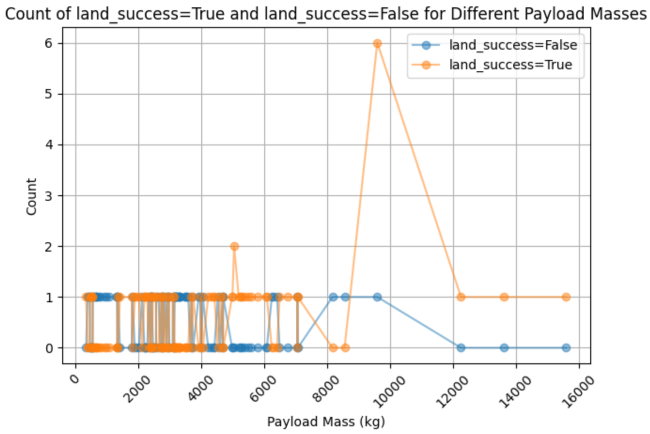


Fig 26: Showing the landing outcomes based on the core used by the Payload mass

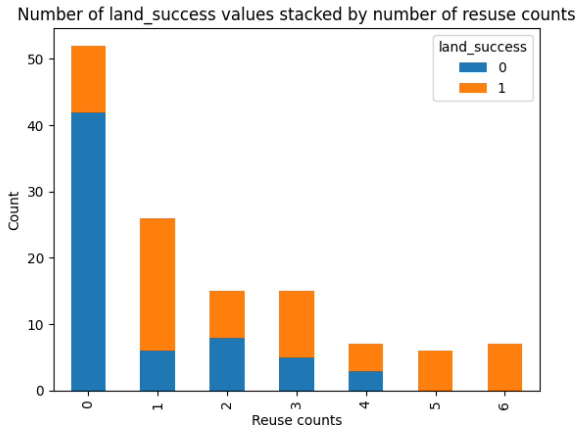


Fig 27: Showing the landing outcomes based on the core used by the number of times a core has been reused

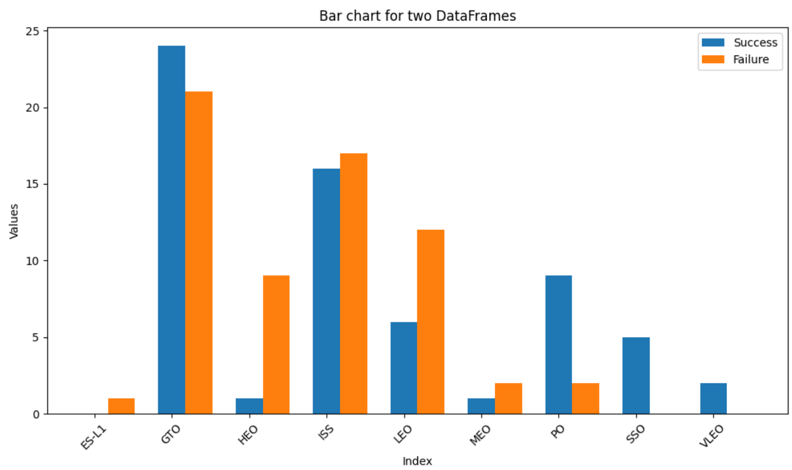


Fig 28: Showing the landing outcomes based on the orbit type the payload was launched into

From the above Figures we can make a few deductions to understand which characteristics of the Falcon 9 have led to the highest number of landing successes. Figure 25 shows that core: 33 has the most number of successes which means that the particular core's specs can be considered ideal and replicated for future missions to increase the landing success rates. Figure 26 shows that the highest number of landing success was observed when the mass of the payload was around 10,000 kgs. This can be considered while planning the mission to increase success rates. Next, Figure 27 shows that the highest success rates belonged to cores that had been reused only once. But cores that have been reused 5-6 times also show good success rates. This tells that the reusability aspect of the cores is a sustainable idea that has shown real life applicability and success. Lastly, Figure 28 shows us that the most orbit types with the highest success rate are the GTO, ISS and LEO orbits. Thus, further study can be done to look into whether a Falcon 9 prefers certain types of orbits over others and why.

6. Conclusion:

This project aimed to enhance the predictability of Falcon 9 first-stage landing outcomes using data science and machine learning techniques. he EDA and Interactive maps provided a strong foundation to understand the various aspects that influence Falcon 9 landings. Whereas, the machine learning aspect was crucial in being able to create a robust predictive model. The Random Forest model proved to be the most accurate with a 94.87% accuracy rate, outperforming other models like XGBoost and KNN. Key features impacting landing success included the core used, payload mass, core reuse frequency, and target orbit type. Notably, cores such as core:33 and payloads around 10,000 kgs showed higher success rates, and reuse counts of 1 and even 5 and 6 showed good success, affirming the sustainability of reusability, and highlighting specific mission planning considerations for the future.

In conclusion, the insights from our analysis provide valuable guidelines for optimizing Falcon 9 missions. By accurately predicting landing outcomes, SpaceX can improve mission reliability and cost-efficiency. The findings suggest that focusing on specific core specifications, optimal payload masses, and understanding the impact of different orbit types can significantly enhance the success rates of future launches. This project successfully emphasizes the potential of machine learning in advancing sustainable and economical space exploration.

References

Bojun, Z., Zhanchao, L., & Gang, L. (2019). High-Precision Adaptive Predictive Entry Guidance for Vertical Rocket Landing. *Journal of Spacecraft and Rockets*, *56*(6), 1735–1741. https://doi.org/10.2514/1.a34450

Elreedy D, A. A., & F., K. (2023). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. Machine Learning. https://doi.org/https://doi.org/10.1007/s10994-022-06296-4

Jones, H. (2018). The Recent Large Reduction in Space Launch Cost. *Ttu-Ir.tdl.org*. https://ttu-ir.tdl.org/handle/2346/74082

Lu, P. (2008). Predictor-Corrector Entry Guidance for Low-Lifting Vehicles. *Journal of Guidance, Control, and Dynamics*, *31*(4), 1067–1075. https://doi.org/10.2514/1.32055

Maheshwari S., J. R., & R.S., J. (2018). An Insight into Rare Class Problem: Analysis and Potential Solutions. Journal of Computer Science, 14 (4), 777–792. https://doi.org/10.3844/jcssp.2018.777.792

Reilly, J. (2024, January 8). *Overfitting and Underfitting in Machine Learning*. Akkio. https://www.akkio.com/post/overfitting-and-underfitting-in-machine-learning#why-does-overfitting-happen

Rich Data Admin. (n.d.). *SMOTE explained for noobs - Synthetic Minority Over-sampling TEchnique line by line · Rich Data*. Rikunert.com. https://rikunert.com/SMOTE_explained

Sippel, M., Sven Stappert, Bussler, L., & Dumont, E. (2017). Systematic Assessment of Reusable First-Stage Return Options. *Proceedings of the International Astronautical Congress, IAC*. https://iafastro.directory/iac/proceedings/IAC-17/

SpaceX-API/docs at master · r-spacex/SpaceX-API. (2020). GitHub. https://github.com/r-spacex/SpaceX-API/tree/master/docs

Zhou, X., & Pham, N. (2023, August). Week 2: Introduction of project [Notes on Blackboard for DATA7901, Slides 8-13 to Nandini Tejas Mehta]. In *Lecture Notes on Blackboard*.