* Include a section discussing the model selection process, feature importance, and model performance.

------------------------------------------------------------------------------------------------------------------

**Model Selection Process:**

For the model selection process, knowing that our problem is a regression problem, I decided to evaluate three different models: Linear Regression, Random Forest Regressor, and XGBoost Regressor.

1. Linear Regression:

* Reason for Selection: I started with a simple linear regression model because it is straightforward to implement and interpret. It serves as a good baseline for comparison.
* Performance: The linear regression model gave the worst performance with an R2 score of -0.503. The negative R2 score indicates that the model indicates that the model performed worse than a horizontal line (mean of the target values). This poor performance suggests that the relationship between the features and the target variable is not linear, and the model fails to capture the complexity of the data.

1. Random Forest Regressor:

* Reason for Selection: Given the limitations of the linear regression model, I chose the Random Forest Regressor as the next model. Random Forest is an ensemble method that can handle non-linear relationships and interactions between features, making it suitable for more complex datasets.
* Performance: The Random Forest Model performed significantly better than the linear regression model, with an R2 score of 0.888. This indicates that the model explains a large portion of the variance in the target variable, capturing the underlying patterns more effectively.

1. XGBoost Regressor:

* Reason for Selection: Finally, I selected the XGBoost Regressor. XGBoost is a powerful gradient boosting algorithm known for its high performance in various regression tasks. It often outperforms other algorithms by effectively handling non-linear relationships and reducing overfitting.
* Performance: The XGBoost model gave an R2 score of 0.866, which is better than the linear regression but slightly worse than the Random Forest Regressor. Despite its high performance, it did not outperform the Random Forest model in this specific case, possibly due to the nature of the dataset or the default hyperparameters used.

**Feature Importance:**

For feature importance, I used Sequential Feature Selector to identify the top 100 features contributing the most to the model’s performance. This step was crucial in reducing the dimensionality of the dataset and improving the efficiency of the models. By focusing on the most relevant features, I ensured that the models were not overfitting to noise and were more interpretable.

**Model Performance:**

Here is a summary of the performance metrics for each model:

* Linear Regression:
  + MSE: 8119850580.675
  + MAE: 25136.067
  + R2 : -0.503
* Random Forest Regressor:
  + MSE: 605876312.086
  + MAE: 17492.275
  + R2 : 0.888

------------------------------------------------------------------------------------------------------------------

* Suggest any potential improvements or next steps for the project with justifying you suggestion.

1. Outlier Detection:

* Identify and handle outliers using techniques like Z-score, IQR, and box plots to improve model performance.

1. Feature Engineering:

* Create new features by exploring feature interactions, creating polynomial features, or leveraging domain knowledge to enhance model accuracy.

1. Hyperparameter Tuning:

* Optimize model hyperparameters using Grid Search to improve model performance.

1. Ensemble Methods:

* Combine multiple models using techniques like bagging, boosting, or stacking to improve overall performance by leveraging their strengths.

1. Cross-Validation:

* Implement K-fold cross-validation to ensure consistent model performance across different data subsets and improve robustness.