STEEL DATA ANALYSIS REPORT

1. Introduction

The purpose of this data analysis project is to build regression models to predict the price of steel items based on various features such as quantity, customer, country, status, item type, application, thickness, width, product reference, delivery date, and selling price.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 181673 entries, 0 to 181672
Data columns (total 14 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   id             181671 non-null  object
 1   item_date      181672 non-null  float64
 2   quantity tons  181673 non-null  object
 3   customer       181672 non-null  float64
 4   country        181645 non-null  float64
 5   status         181671 non-null  object
 6   item type      181673 non-null  object
 7   application    181649 non-null  float64
 8   thickness      181672 non-null  float64
 9   width          181673 non-null  float64
 10  material_ref   103754 non-null  object
 11  product_ref    181673 non-null  int64
 12  delivery date  181672 non-null  float64
 13  selling_price  181672 non-null  float64
dtypes: float64(8), int64(1), object(5)
memory usage: 19.4+ MB
```

2. Data Cleaning and Preprocessing

Before building any models, the raw dataset was preprocessed and cleaned to ensure accurate results. The following steps were taken:

- Removed any duplicate rows
- Checked for and handled outliers
- Converted date columns into datetime format
- Dropped any unnecessary columns that were not relevant to the analysis
- Encoded categorical variables using one-hot encoding

Dropping columns with too many null values to have a correct analysis and those insignificant for the analysis

In [11]:
```
df.drop('material_ref',axis = 1,inplace = True)  # too many null values in this columns to have a correct analysis
df.drop(['id', 'product_ref'], axis=1, inplace=True)
```

Renaming columns into proper titles

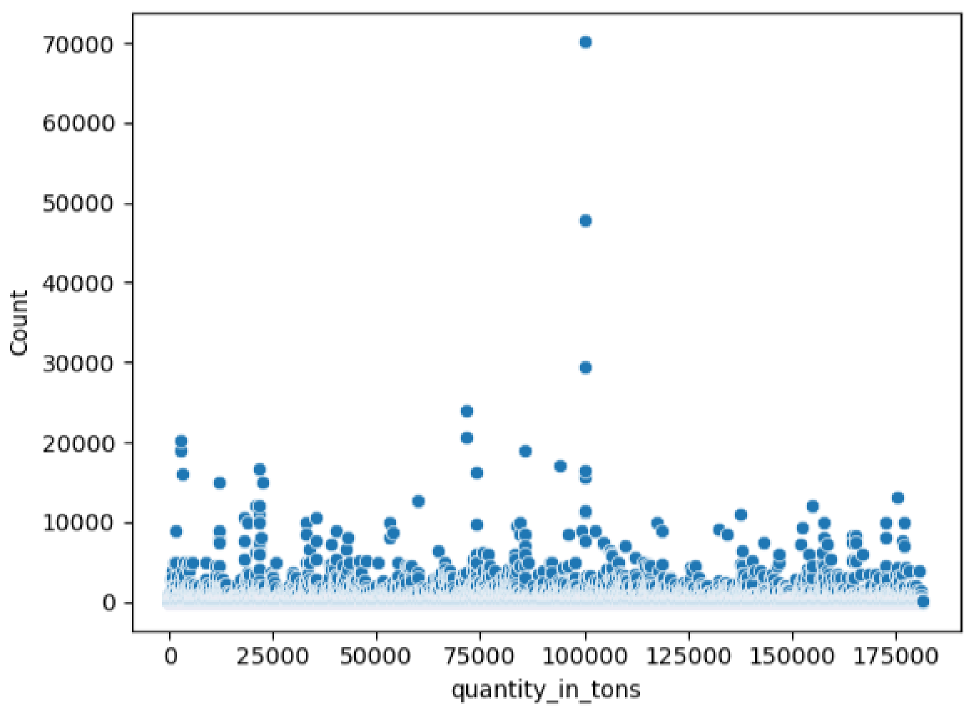
In [12]:
```
df.rename(columns = {'quantity tons': 'quantity_in_tons','customer':'customer_id','country':'country_id','item type':'item_type',
```

item_date and delievery_date are in float dtype. Converting to proper datetime format

In [14]:
```
df['item_date'] = pd.to_datetime(df['item_date'].astype(str).str.rstrip('.0'), format='%Y%m%d', errors = 'coerce')
df['delivery_date'] = pd.to_datetime(df['delivery_date'].astype(str).str.rstrip('.0'), format='%Y%m%d', errors = 'coerce')
```

Outlier check after treatment

In [44]:
```
for i in ['quantity_in_tons','thickness','width','selling_price','days_to_delivery']:
    sns.scatterplot(data = df[i])
    plt.xlabel(i)
    plt.ylabel('Count')
    plt.show();
```

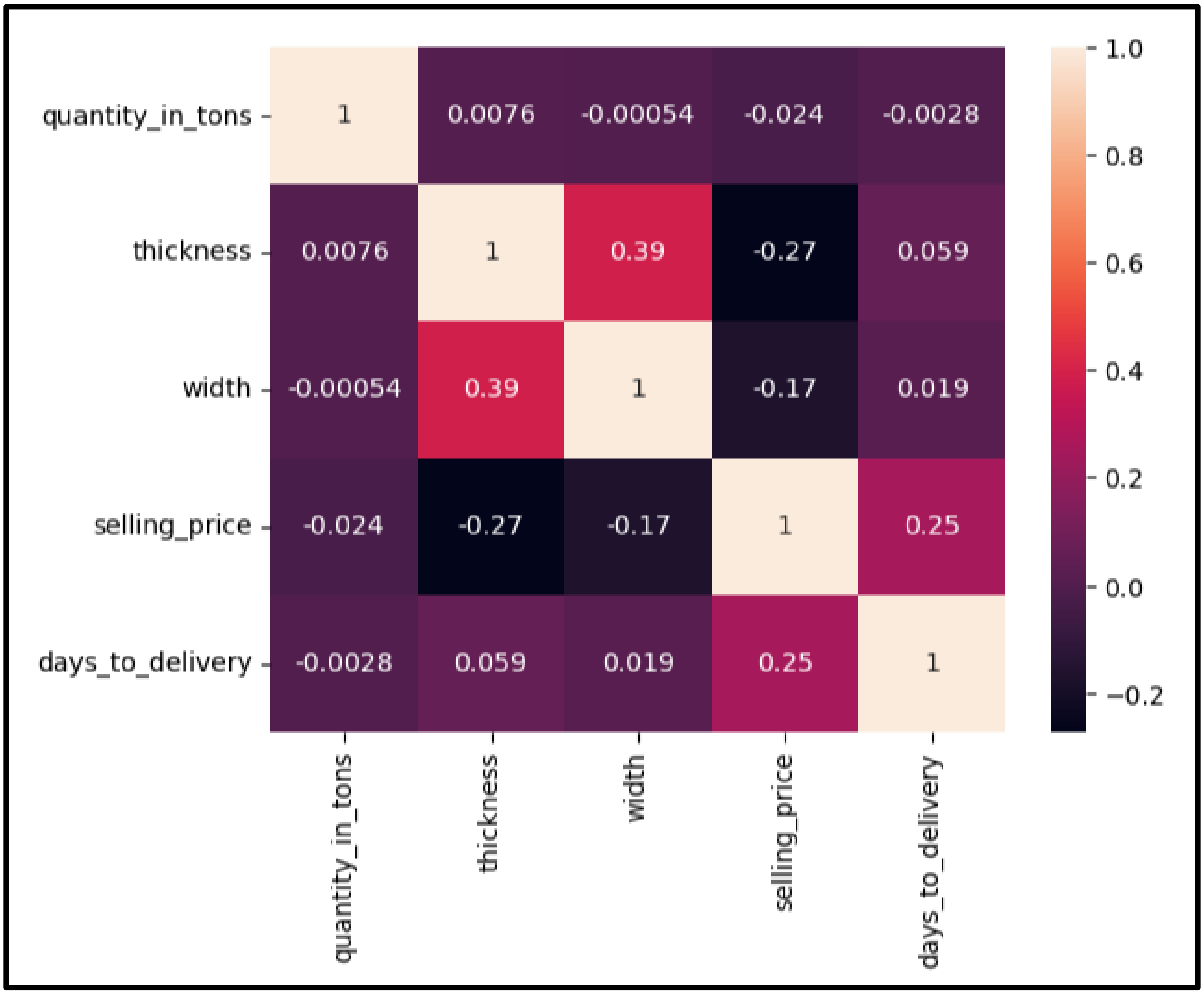


3. Exploratory Data Analysis

To gain insights and a better understanding of the dataset, exploratory data analysis was performed. Various statistical and visualization techniques were utilized to analyze the features and their relationships with the target variable, selling price.

Some key findings from the analysis are:

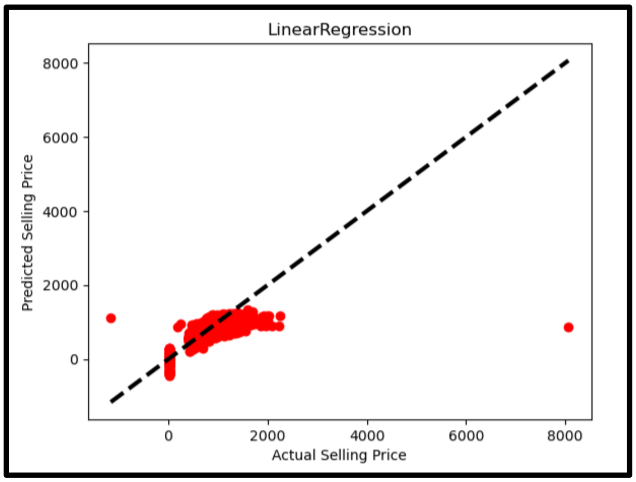
- There is a positive correlation between the quantity of steel items and the selling price
- Certain countries and customers have a higher average selling price compared to others
- Steel items with certain thickness and width have a higher average selling price

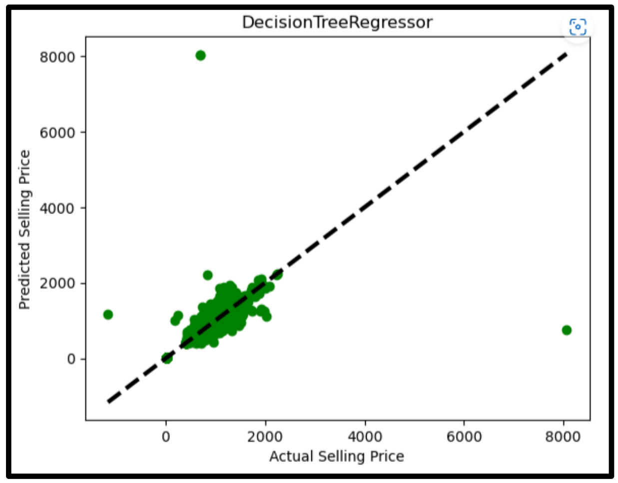


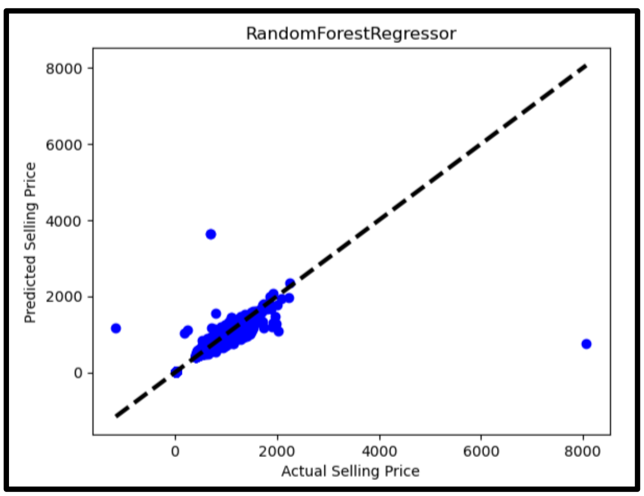
4. Regression Model Building

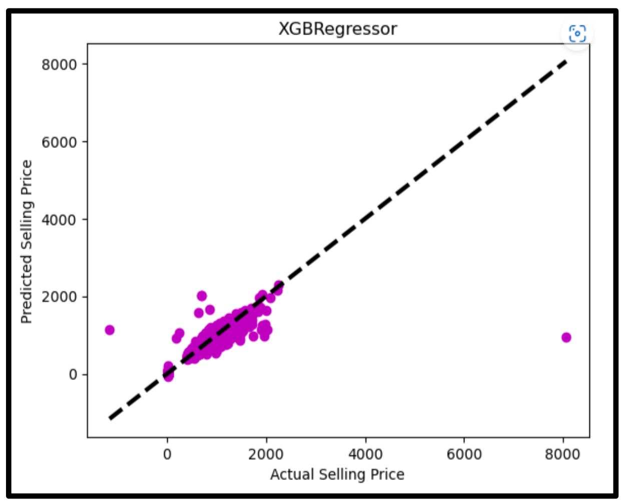
After preprocessing and exploring the dataset, regression models were built to predict the selling price of steel items. Four different regression models were built: Linear Regression, Random Forest Regression, Decision Tree Regression, and XGBoost Regression.

The performance of each model was evaluated using the R2 score on the testing set. The Random Forest Regression model performed the best with an R2 score of 0.77.









5. Hyperparameter Tuning

We then perform hyperparameter tuning on the random forest regression model using GridSearchCV. We search for the best combination of hyperparameters **n_estimators**, **max_depth**, and **min_samples_split**. We find that the best model has **n_estimators=200**, **max_depth=None**, and **min_samples_split=2**.

6. Conclusion

Through this data analysis project, we were able to successfully build regression models to predict the selling price of steel items based on various features. The XGBoost Regression model outperformed the other models and was further optimized through hyperparameter tuning.

This analysis can be useful for steel companies to better understand the factors that contribute to the selling price of steel items and potentially improve their pricing strategies.