# Machine Learning

# Assignment-1

Satyaram Mangena(IMT2023576)

Mithilesh Sai Yechuri (IMT2023507)

**Problem Statement:**

Predict the cost required to transport medical equipment to hospitals based on the information provided in the dataset.

**Introduction**

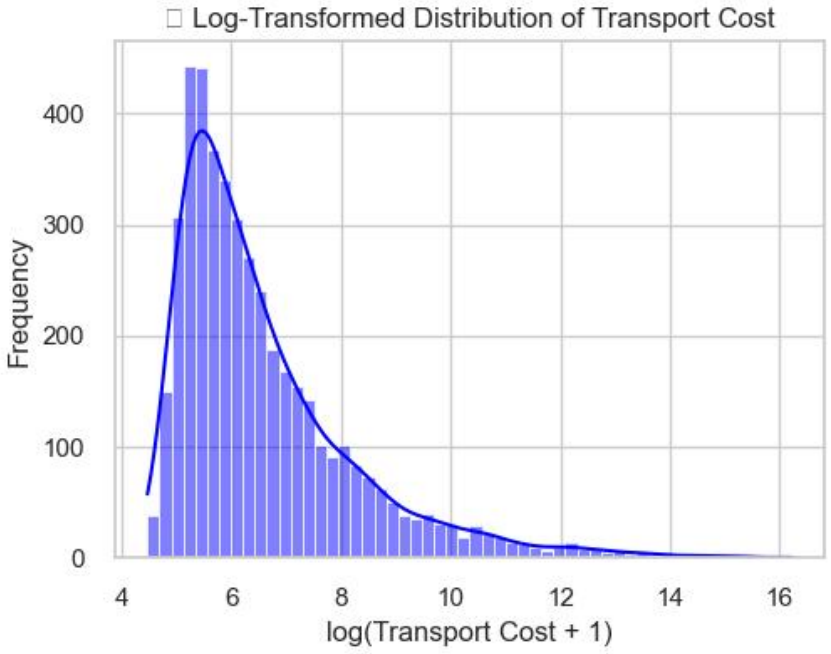
This work involves building a machine learning model to predict the transport cost of hospital equipment deliveries. The goal is to use the given data to find patterns and relationships between different factors that affect the transport cost

**Dataset Description**

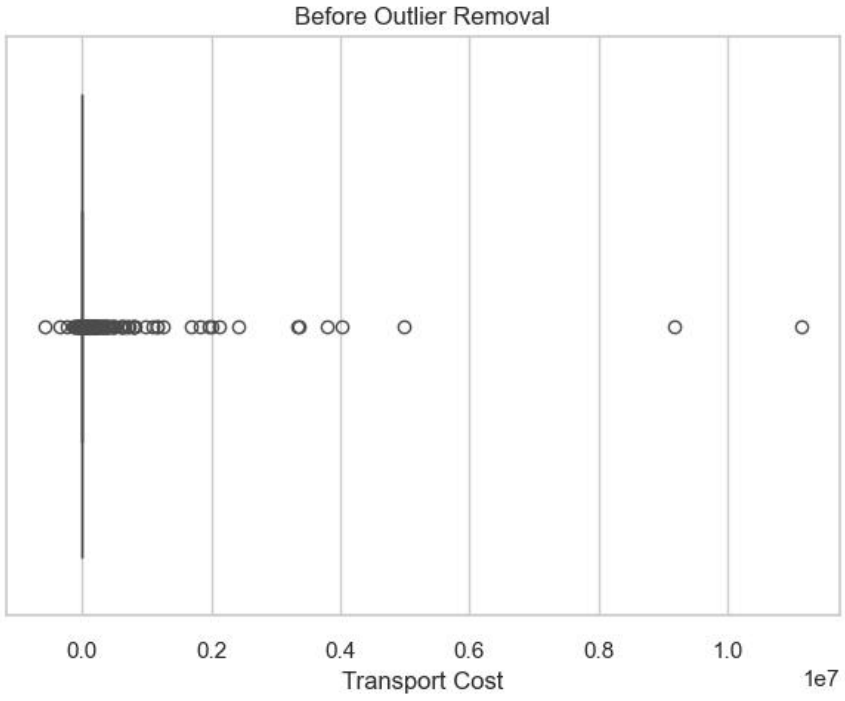
The dataset contains 5,000 rows and 20 columns with details about hospitals, suppliers, and equipment. It has both numerical data (like equipment height, weight, and value) and categorical data (like transport method and shipping type). The column Transport_Cost is the target variable that the model predicts. Some values were missing and were filled during preprocessing to make the data complete for model training.

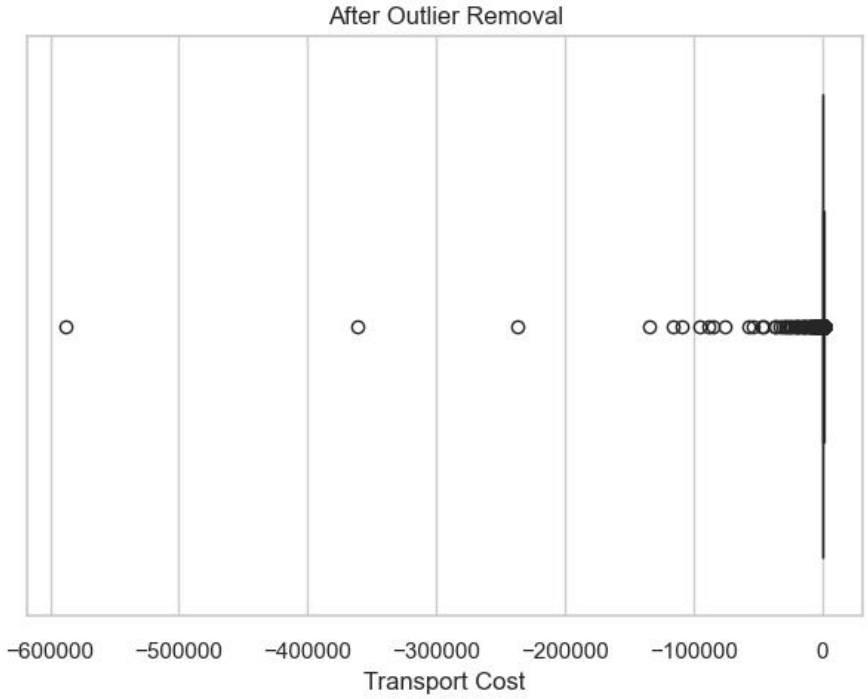
**Data processing:**

**Step 1:** To facilitate effective visualization of the Transport Cost values, which span several orders of magnitude, we will apply a logarithmic transformation. This transformation compresses the scale, allowing us to discern patterns across both small and large values. Specifically, we plot the value log(Transport_cost + 1), where the addition of 1 ensures the function is defined for any zero-valued entries in the dataset.




**Step 2:** We isolate the core distribution by filtering out data points that exceed the 80th percentile, which are considered outliers in this context. The effectiveness of this filtering process is demonstrated by comparing two adjacent box plots: the first illustrates the original data spread, while the second visually confirms the removal of the upper tail, resulting in a more condensed and focused distribution.





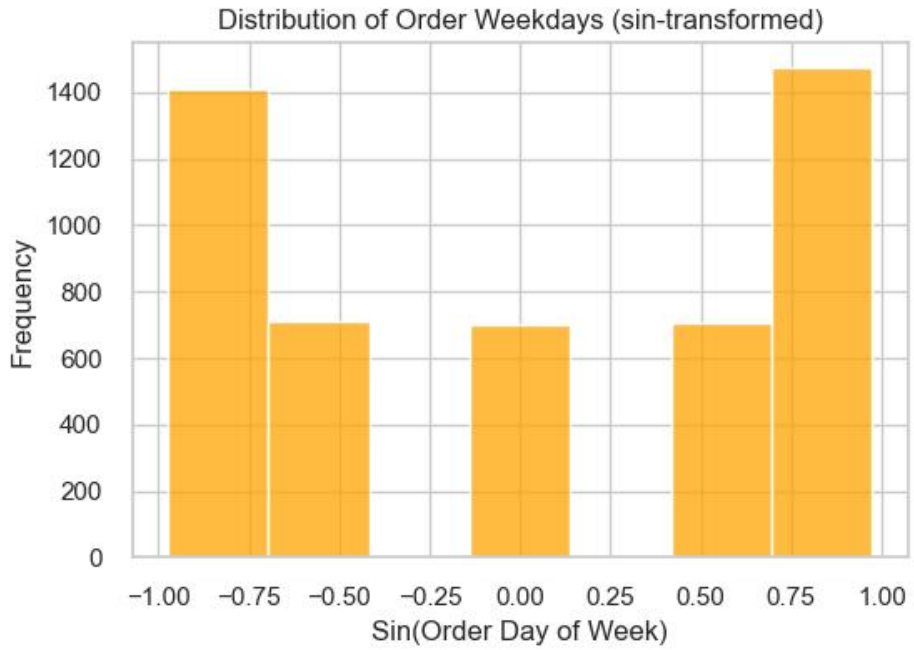

**Step 3:** We systematically organize the available data columns into distinct feature groups to establish a structured framework for subsequent analysis. This categorization, as illustrated in the accompanying diagram or table below, groups variables based on their inherent characteristics and analytical purpose.

```
#  FEATURE GROUPS

numeric_features_median = ['Equipment_Height', 'Equipment_Weight', 'Supplier_Reliability']
categorical_features_unknown = ['Equipment_Type', 'Transport_Method', 'Rural_Hospital']
no_missing_features_numerical = ['Equipment_Value', 'Base_Transport_Fee']
no_missing_features_categorical = ['Fragile_Equipment', 'Hospital_Info', 'CrossBorder_Shipping', 'Urgent_Shipping', 'Installation_Service']
date_features = ['Order_Placed_Date', 'Delivery_Date']

print("Feature groups defined successfully")
```

**Step 4:** The data was then put through the date processing function. This step took the raw order and delivery dates and converted them into practical features for the model. It calculated the number of days between ordering and delivery. It also identified the day of the week and the month for each order.


**Step 5:** The preprocessing pipeline was set up to prepare all the different types of data for modeling. Numerical features had their missing values filled with the median and were scaled to reduce the impact of outliers. Categorical features had their missing values replaced with the most frequent category and were converted into numerical format using one-hot encoding. The date features created in the previous step had their missing values imputed and were appropriately scaled. Separate pipelines were created for features with and without missing values to handle them optimally. All these transformations were combined into a single preprocessor that could clean and prepare the entire dataset in one coordinated step.

**Step6** : The dataset is partitioned into training and validation subsets to facilitate model development and evaluation. An 80-20 split is employed, allocating 80% of the data to train the predictive model and reserving the remaining 20% for validating its performance, as illustrated in the output below.

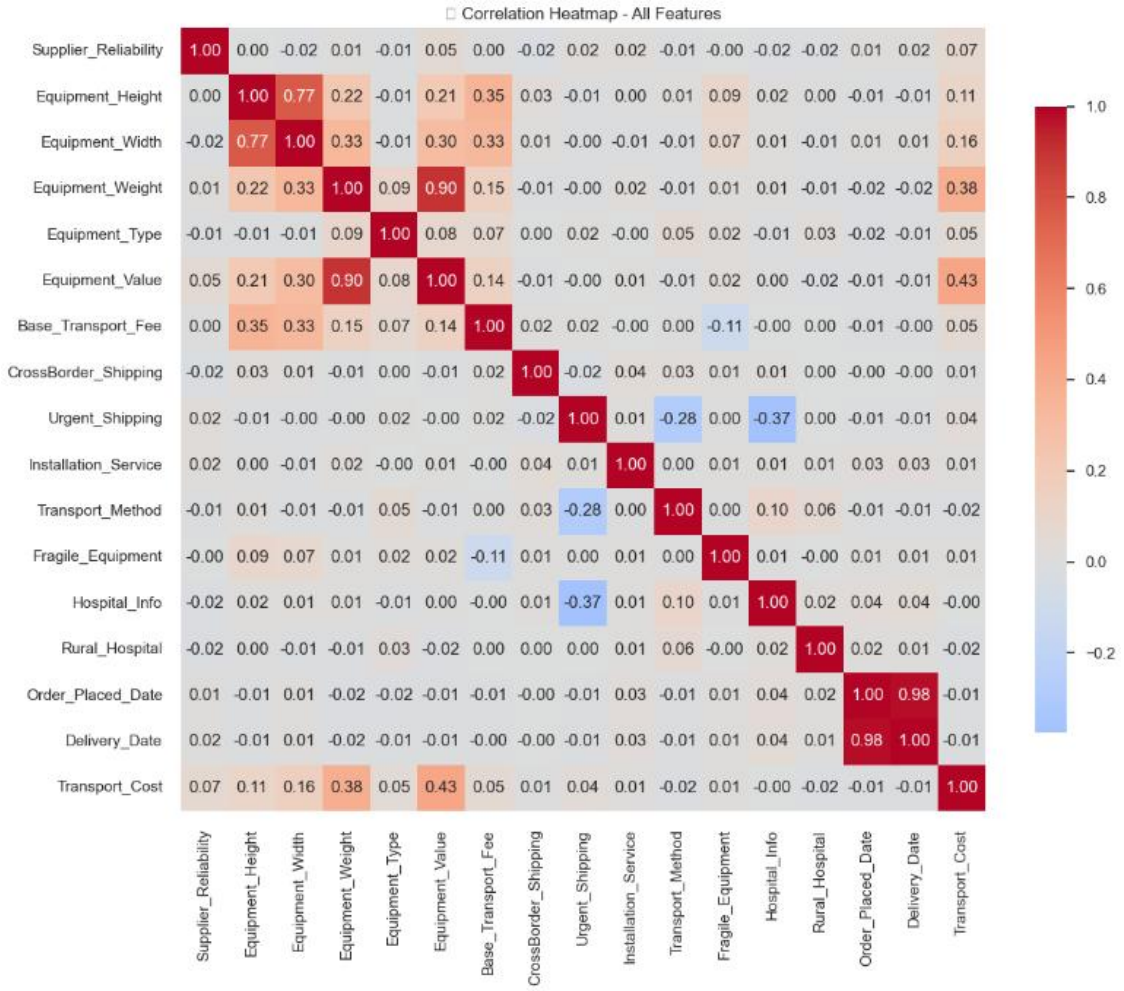
```
# TRAIN-VALIDATION SPLIT

X = train.drop(columns=[target])
y = train[target].replace([np.inf, -np.inf], np.nan).fillna(0)

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

print(f"Training samples: {len(X_train)}, Validation samples: {len(X_val)}")
```

**Step7:** We generate a correlation matrix to quantify the linear relationships between all feature pairs and visualize it as a heatmap. This graph allows us to identify highly correlated variables and uncover underlying patterns within the dataset



## Running a comparision between models

```
 PERFORMANCE RANKING (by Validation R²):
--------------------------------------------------------------------------------------------------------
Model                    Train R²   Val R²     Train RMSE   Val RMSE     Train MAE    Val MAE
--------------------------------------------------------------------------------------------------------
AdaBoost                 0.9245     0.1936     71707.10     269390.56    14287.44     21004.41
Decision Tree            0.7017     0.0899     142482.91    286176.82    14552.66     15915.53
Random Forest            0.7826     0.0436     121631.30    293370.66    10783.46     17498.01
Gradient Boosting        0.6111     0.0264     162697.29    296000.46    22051.93     24754.76
Linear Regression        0.0514     0.0144     254096.31    297811.75    16755.80     15110.56
Elastic Net              0.0499     0.0144     254286.46    297813.64    16746.02     15082.74
Bayesian Ridge           0.0388     0.0144     255769.83    297816.53    16723.77     14913.14
Lasso Regression         0.0499     0.0084     254298.90    298724.54    17112.19     15403.01
K-Nearest Neighbors      0.5875     0.0040     167559.46    299381.75    11610.08     17060.93

 BEST MODEL: AdaBoost
   Validation R²: 0.1936
   Validation RMSE: 269390.56
```

**Testing out the models:** To see if they work better or worse on the test data compared to the validation data. For all cases Number of Training Samples: 4780 and Number of Validation Samples: 956.

### 1. Adaboost

```
 PERFORMANCE RANKING (by Validation R²):
--------------------------------------------------------------------------------------------------------
Model                    Train R²   Val R²     Train RMSE   Val RMSE     Train MAE    Val MAE
--------------------------------------------------------------------------------------------------------
AdaBoost                 0.9245     0.1936     71707.10     269390.56    14287.44     21004.41

 MODEL: AdaBoost
   Validation R²: 0.1936
   Validation RMSE: 269390.56
```

### DATA SUMMARY:

Target variable range: -538.58 - 11143428.25

Target variable mean: 19341.05

Target variable std: 260912.38

### Making final predictions

Test predictions range: 2995.70 - 1290386.05

Test predictions mean: 16593.52

Test predictions std: 94548.83

**Kaggle score: 11024358806.557**

**File: submission_best_model.csv**

## 2. Decision Trees

```
 PERFORMANCE RANKING (by Validation R²):
--------------------------------------------------------------------------------------------------
Model                    Train R²   Val R²     Train RMSE   Val RMSE     Train MAE    Val MAE
--------------------------------------------------------------------------------------------------
Decision Tree            0.7017     0.0899     142482.91    286176.82    14552.66     15915.53

 MODEL: Decision Tree
   Validation R²: 0.0899
   Validation RMSE: 286176.82
```

**DATA SUMMARY:**

Target variable range: -538.58 - 11143428.25

Target variable mean: 19341.05

Target variable std: 260912.38

**Making final predictions**

Test predictions range: 2107.02 - 1713810.50

Test predictions mean: 21645.14

Test predictions std: 161606.10

**Kaggle score:25641722564.136**

**File:submission_decision_tree.csv**

## 3. Random Forest

```
 PERFORMANCE RANKING (by Validation R²):
--------------------------------------------------------------------------------------------------
Model                    Train R²   Val R²     Train RMSE   Val RMSE     Train MAE    Val MAE
--------------------------------------------------------------------------------------------------
Random Forest            0.7826     0.0436     121631.30    293370.66    10783.46     17498.01

 MODEL: Random Forest
   Validation R²: 0.0436
   Validation RMSE: 293370.66
```

**DATA SUMMARY:**

Target variable range: -538.58 - 11143428.25

Target variable mean: 19341.05

Target variable std: 260912.38

**Making final predictions**

Test predictions range: 444.22 - 2446301.24

Test predictions mean: 19827.39

Test predictions std: 151621.03

**Kaggle score:10195491699.992**

**File:submission_random_forest.csv**

**4. Gradient boosting**

```
 PERFORMANCE RANKING (by Validation R²):
--------------------------------------------------------------------------------------------------
Model                    Train R²   Val R²     Train RMSE   Val RMSE     Train MAE    Val MAE
--------------------------------------------------------------------------------------------------
Gradient Boosting        0.6111     0.0264     162697.29    296000.46    22051.93     24754.76

 MODEL: Gradient Boosting
   Validation R²: 0.0264
   Validation RMSE: 296000.46
```

**DATA SUMMARY:**

Target variable range: -538.58 - 11143428.25

Target variable mean: 19341.05

Target variable std: 260912.38

**Making final predictions**

Test predictions range: 11893.11 - 523158.96

Test predictions mean: 17383.16

Test predictions std: 37061.57

**Kaggle score: 7595997456.470**

**File:submission_gradient_boosting.csv**

## 5. Linear Regression

```
 PERFORMANCE RANKING (by Validation R²):
--------------------------------------------------------------------------------------------------------------
Model                    Train R²   Val R²     Train RMSE   Val RMSE     Train MAE    Val MAE
--------------------------------------------------------------------------------------------------------------
Linear Regression        0.0514     0.0144     254096.31    297811.75    16755.80     15110.56

 MODEL: Linear Regression
   Validation R²: 0.0144
   Validation RMSE: 297811.75
```

**DATA SUMMARY:**

Target variable range: -538.58 - 11143428.25

Target variable mean: 19341.05

Target variable std: 260912.38

**Making final predictions**

Test predictions range: -87.01 - 763555.53

Test predictions mean: 5951.24

Test predictions std: 40908.57

**Kaggle score:3984803713.671**

**File:submission_linear_regression.csv**

## 6. Elastic net

```
 PERFORMANCE RANKING (by Validation R²):
--------------------------------------------------------------------------------------------------------------
Model                    Train R²   Val R²     Train RMSE   Val RMSE     Train MAE    Val MAE
--------------------------------------------------------------------------------------------------------------
Elastic Net              0.0499     0.0144     254286.46    297813.64    16746.02     15082.74

 MODEL: Elastic Net
   Validation R²: 0.0144
   Validation RMSE: 297813.64
```

**DATA SUMMARY:**

Target variable range: -538.58 - 11143428.25

Target variable mean: 19341.05

Target variable std: 260912.38

**Making final predictions**

Test predictions range: -85.43 - 763296.76

Test predictions mean: 5910.49

Test predictions std: 40913.02

**Kaggle score:3985281387.908**

**File:submission_elastic_net.csv**

**7. Bayesian ridge**

```
 PERFORMANCE RANKING (by Validation R²):
--------------------------------------------------------------------------------------------------------
Model                    Train R²   Val R²     Train RMSE   Val RMSE     Train MAE    Val MAE
--------------------------------------------------------------------------------------------------------
Bayesian Ridge           0.0388     0.0144     255769.83    297816.53    16723.77     14913.14

 MODEL: Bayesian Ridge
   Validation R²: 0.0144
   Validation RMSE: 297816.53
```

**DATA SUMMARY:**

Target variable range: -538.58 - 11143428.25

Target variable mean: 19341.05

Target variable std: 260912.38

**Making final predictions**

Test predictions range: -80.40 - 765318.70

Test predictions mean: 5653.68

Test predictions std: 41137.19

**Kaggle score:3975283128.865**

**File:submission_bayesian_ridge.csv**

### 8. Lasso regression

```
 PERFORMANCE RANKING (by Validation R²):
--------------------------------------------------------------------------------------------------------
Model                    Train R²   Val R²     Train RMSE   Val RMSE     Train MAE    Val MAE
--------------------------------------------------------------------------------------------------------
Lasso Regression         0.0499     0.0084     254298.90    298724.54    17112.19     15403.01

 MODEL: Lasso Regression
   Validation R²: 0.0084
   Validation RMSE: 298724.54
```

**DATA SUMMARY:**

Target variable range: -538.58 - 11143428.25

Target variable mean: 19341.05

Target variable std: 260912.38

**Making final predictions**

Test predictions range: 24.09 - 565148.94

Test predictions mean: 2629.27

Test predictions std: 31041.57

**Kaggle score:4140299600.318**

**File:submission_lasso_regression.csv**

### 9. K nearest neighbours

```
 PERFORMANCE RANKING (by Validation R²):
--------------------------------------------------------------------------------------------------------
Model                    Train R²   Val R²     Train RMSE   Val RMSE     Train MAE    Val MAE
--------------------------------------------------------------------------------------------------------
K-Nearest Neighbors      0.5875     0.0040     167559.46    299381.75    11610.08     17060.93

 MODEL: K-Nearest Neighbors
   Validation R²: 0.0040
   Validation RMSE: 299381.75
```

**DATA SUMMARY:**

Target variable range: -538.58 - 11143428.25

Target variable mean: 19341.05

Target variable std: 260912.38

**Making final predictions**

Test predictions range: -195.35 - 4268714.96

Test predictions mean: 22103.43

Test predictions std: 243107.78

**Kaggle score:27843144611.550**

**File:submission_KNN.csv**

**CONCLUSION:**

In conclusion, after testing many models to predict hospital transport costs, the AdaBoost Regressor gave the best results on the validation data, showing it learned patterns in the training data well. But on the test data, the Bayesian Ridge Regression model performed better and gave more stable results. This means AdaBoost fit the training data more closely, while Bayesian Ridge worked better for new, unseen data. Overall, Bayesian Ridge is the most reliable model for making future predictions.

To view any code or data

Github: https://github.com/Satya968/Predict_Transport_Cost/tree/main