Supply Chain Shipment Pricing Prediction for HIV/AIDS Commodities

ADTA 5340 Discovery and Learning with Big Data

Team :
Sai Swetha Annem
Azmath Noorain
Srilakshmi Savithena

*This image is generated using AI.

Problem Statement

- The global fight against HIV/AIDS has been going on for decades.
- Organizations across the globe are spending billions on HIV related shipments.
- Current challenges include:
  - Understanding factors influencing shipment costs.
  - Identifying why cost variations occur.
  - Improving shipment efficiency and predictability.
- This project aims to analyze pricing trends, shipment patterns, and predict costs.
- Leveraging machine learning for accurate cost forecasting and decision-making

*This image is generated using AI.

Business Understanding : Importance of Predicting Freight Cost

- Every year around **$5 billion (USD)** is being spent over procuring and supplying HIV related shipments.
- Predicting the freight cost prices would have **organizations to allocate budget effectively.**
- It also helps **choosing better vendors** across the world.
- Forecasting freight cost will **ensure better resource allocation.**

Costs Optimization

Enhanced Operations

Improved Decision Making

*These images are generated using AI.

Data Understanding

The data has been collected from https://catalog.data.gov/. It contains contains commodity pricing and supply chain expenses similar to Global Fund organization

10000
Records

30
Variables

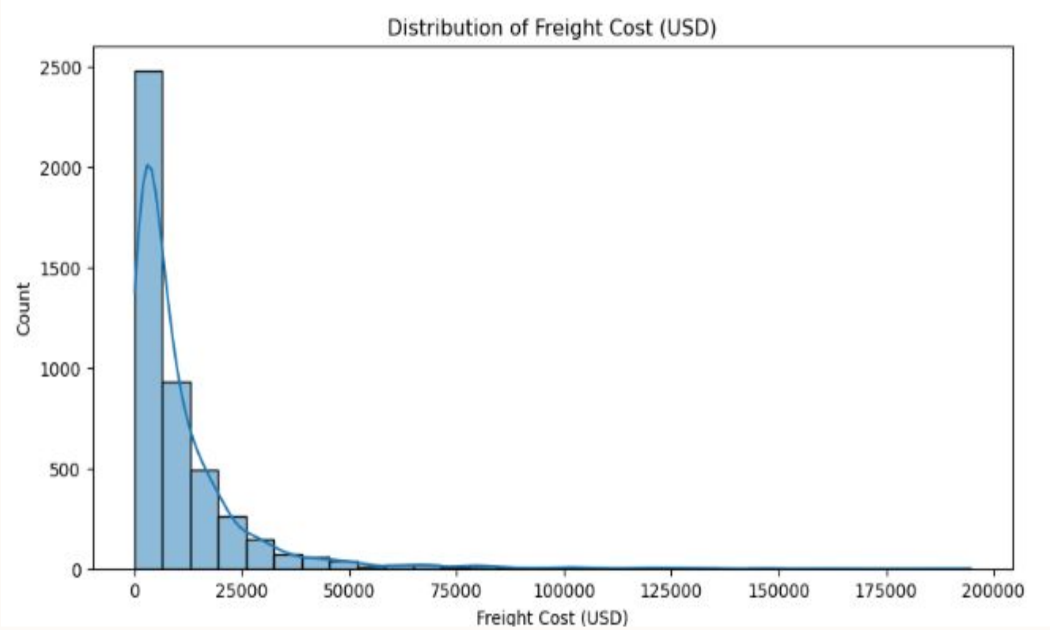
43
Countries

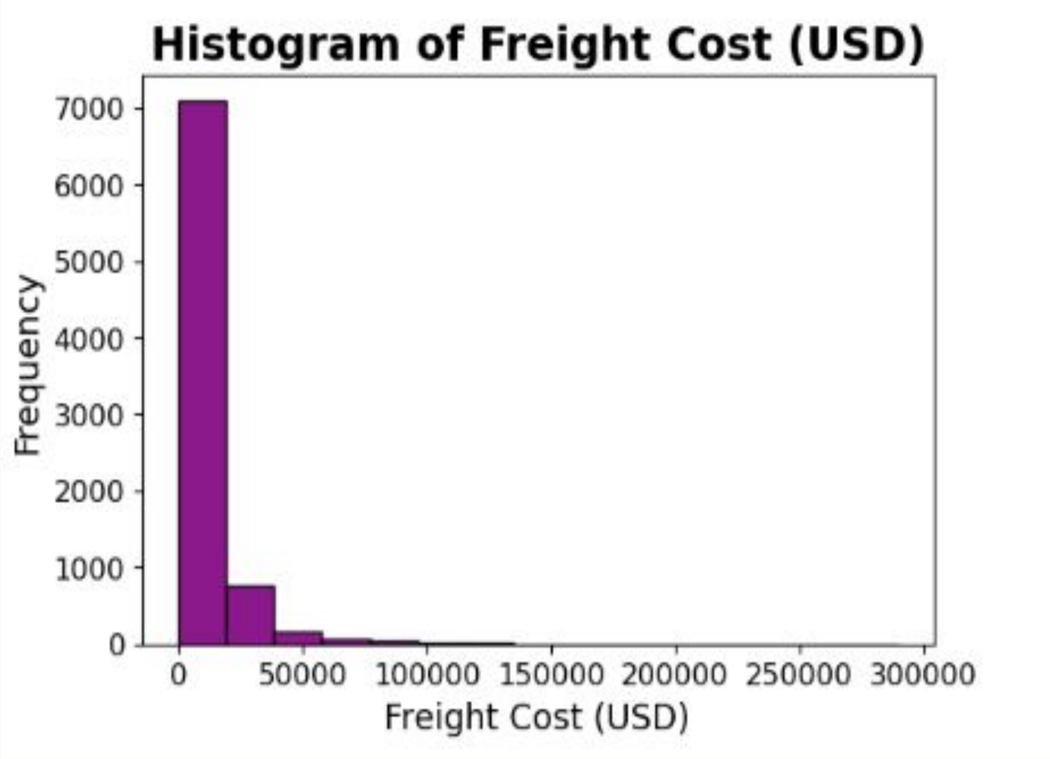
37
Manufacturing Sites

- The freight cost for around 15% of the records was missing , so the data has been removed.
- For certain variables like weight (kilograms) , some of the values were missing , they are replaced with mean imputation

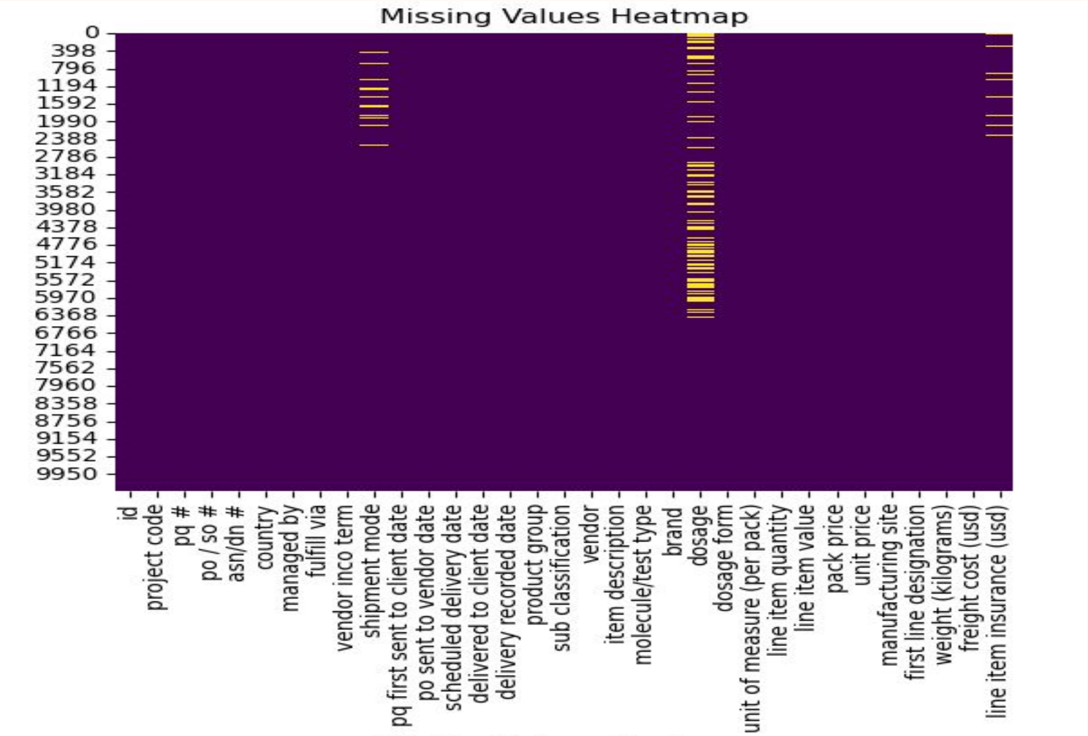
*These images are generated using AI.

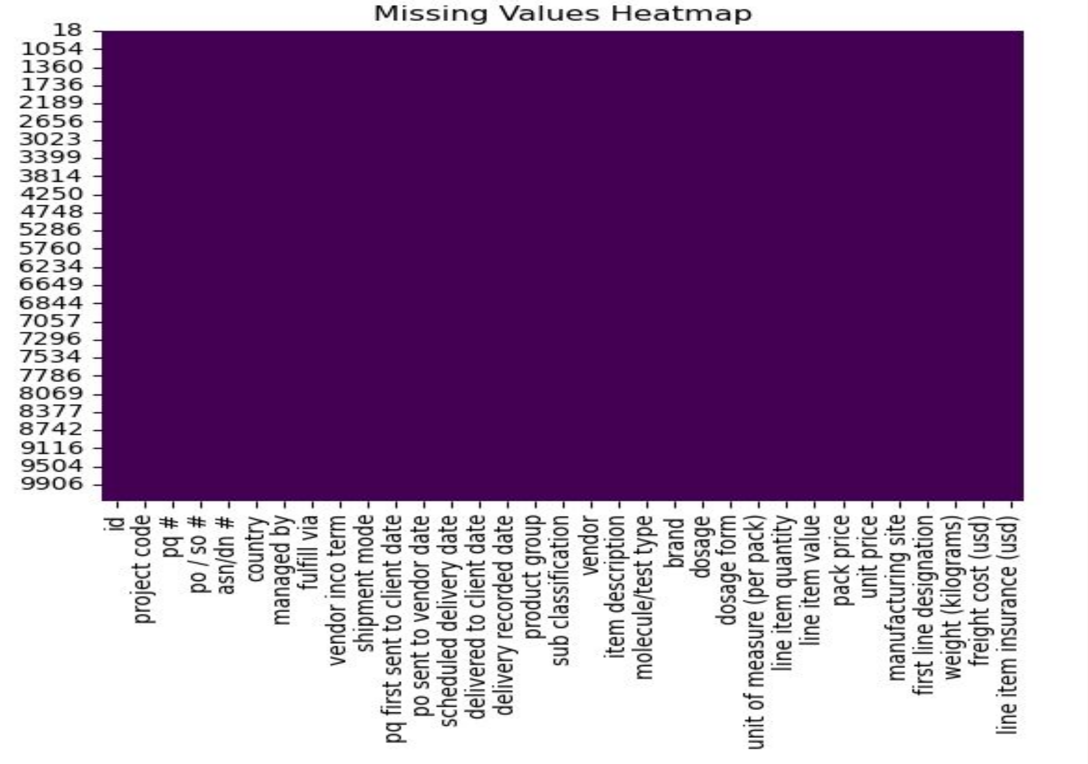
Data Understanding





Data Operations





WorkFlow

1

Exploratory Data Analysis

Analyze Past Shipment to uncover freight cost trends

2

Data Transformation/Preparation

Data Transformation is done before running the models

3

Modelling and Predictions

Ml Models are used for making predictions on unseen data

4

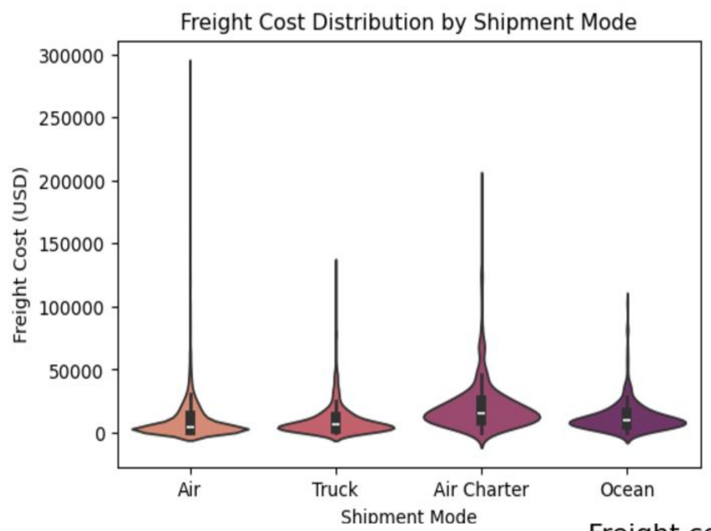
Evaluation

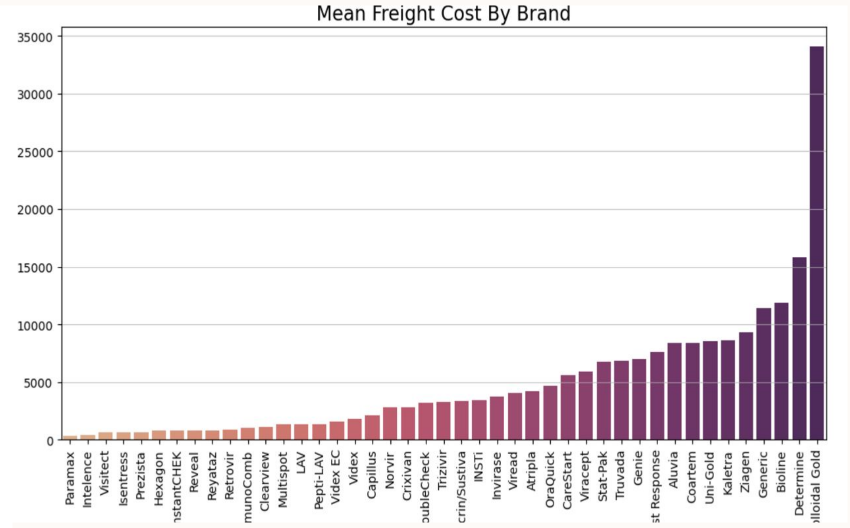
Both the models are evaluated using different metrics

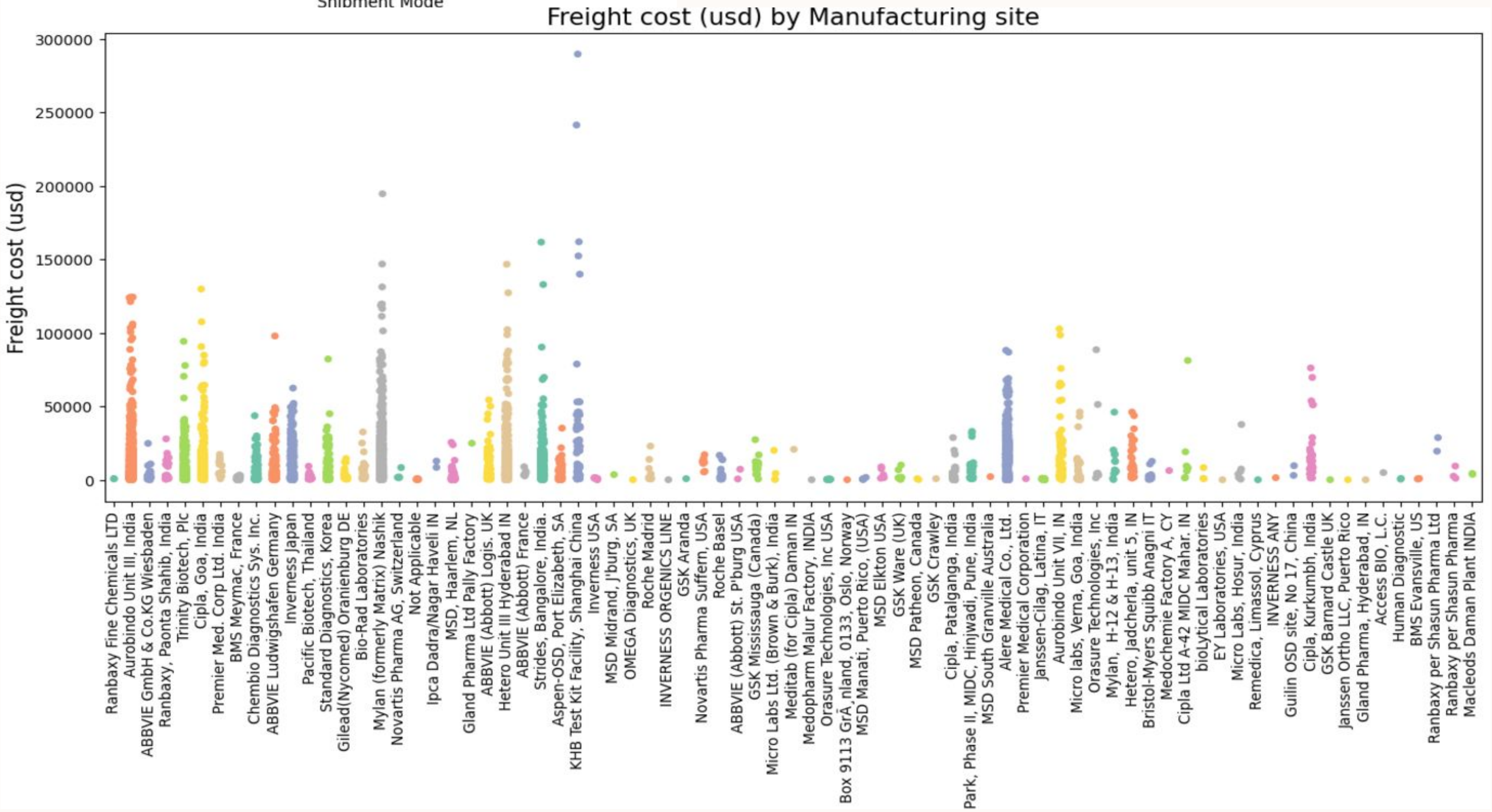
*This image is generated using AI.

Exploratory Data Analysis

- The dataset has been segregated into numerical and categorical features.
- The value of freight cost ranges from **$50 to $300000 dollars.**
- The **shipment mode Air** has higher freight cost compared to others.
- The Injections and test kits have higher costs compared to other medical supplies
- The key categorical features that have shown influence on freight cost are Shipment Mode, Brand , Manufacturing Site , Dosage, Dosage Form , Vendor, Shipment Mode, Product Group.







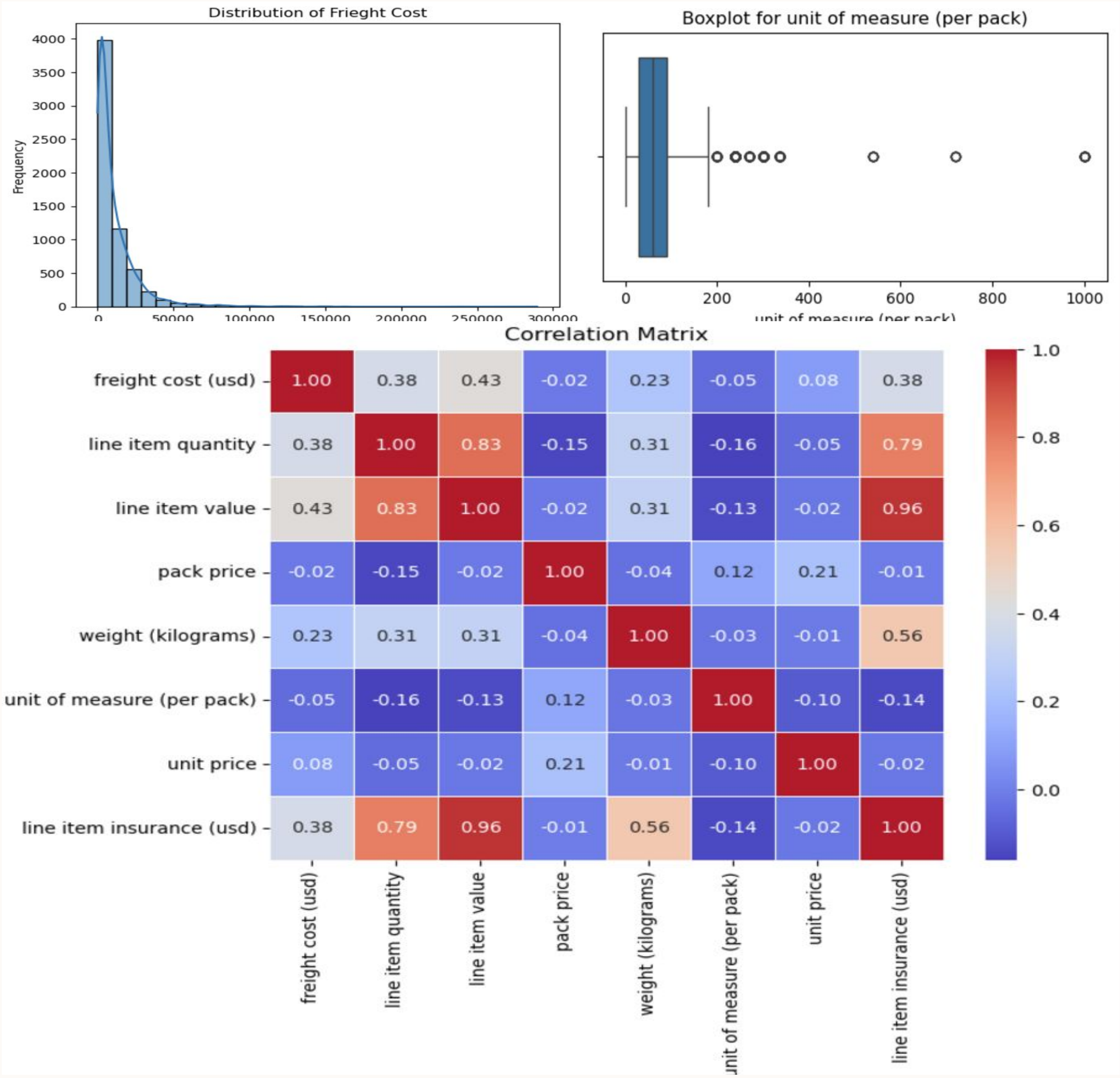
Exploratory Data Analysis

- From the correlation matrix we can understand that there is some correlation between **weight, line item insurance, line item quality and line item value .**

- But there is **no strong correlation** between freight cost and other features.

- Also there is very strong correlation between line item value and line item insurance.

- while the line item quantity and weight has moderate correlation.



Data Transformation

One Hot Encoding :

The categorical features are converted into numeric values using One Hot Encoding

```
categorical_cols = ['country',  'dosage form','dosage','manufacturing site','brand','managed by', 'fulfill via', 'vendor
                    'shipment mode', 'product group', 'sub classification',
                        'vendor','first line designation',
                    'molecule/test type']

# One-hot encoding with drop_first=True to avoid multicollinearity
df_encoded = pd.get_dummies(df, columns=categorical_cols, drop_first=True)
```

After performing one hot encoding we would be having 397 features.

Data Splitting and Standardization:

The data has been split into train and test data. With 80% of training set and 20% of test data and we would be performing standardization on all the values.

```
X = df_encoded.drop('freight cost (usd)', axis=1)  # Drop target column to get features
y = df_encoded['freight cost (usd)']
```

```
# Split data into training (80%) and testing (20%) sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.preprocessing import StandardScaler

# Standardization
scaler = StandardScaler()
X_standardized = scaler.fit_transform(X_train)

print("Standardized Features:")
print(X_standardized)
```

Modelling

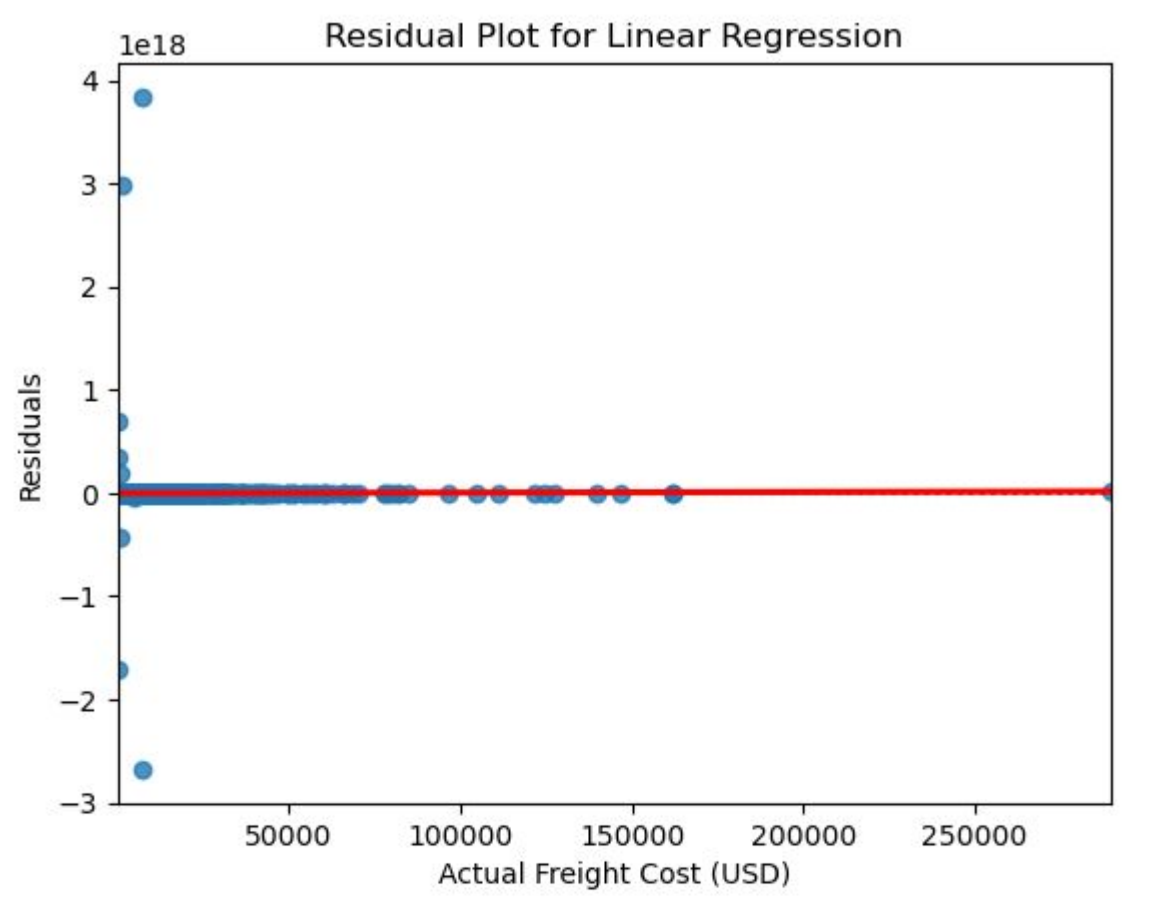
Model 1: Linear Regression

```
# Step 5: Model 1 - Linear Regression
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)  # Train the model

# Make predictions
lr_preds = lr_model.predict(X_test)

# Evaluate the model
lr_rmse = mean_squared_error(y_test, lr_preds, squared=False)
lr_mae = mean_absolute_error(y_test, lr_preds)
lr_r2 = r2_score(y_test, lr_preds)

print("Linear Regression RMSE:", lr_rmse)
print("Linear Regression MAE:", lr_mae)
print("Linear Regression R²:", lr_r2)
```



Model Evaluation:

RMSE - 1.4592384374343834e+17
MAE - 7945422735559329.0
R^2 - -8.29507842672158e+25

From R^2 , we can understand that only 25% percentage of the data is being captured by the model.

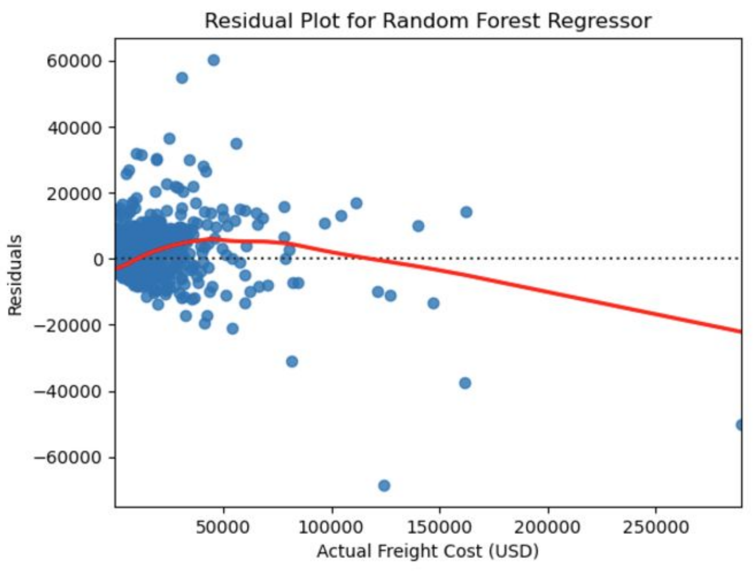
Modelling

Model 1: Random Forest Regressor

```python
# Predict on the test set
y_pred = rf_model.predict(X_test)

# Calculate evaluation metrics
mse = mean_squared_error(y_test, y_pred,squared=False)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R-squared:", r2)
```



Model Evaluation:

RMSE - 8792.26

R^2 - 0.70

From R^2 , we can understand that only 70% percentage of the data is being captured by the model.

Predictions on Unseen Data

Linear Regression Model

```
sample_df = pd.DataFrame([sample_record])

# Ensure columns match the model's training data
sample_df = sample_df.reindex(columns=X_train.columns, fill_value=0)

# Make prediction
prediction = rf_model.predict(sample_df)
print("Predicted Freight Cost (USD):", prediction[0])
```

Predicted Freight Cost (USD): 11013.656177404771

```
sample_record = {
    'unit of measure (per pack)': 100,
    'line item quantity': 25,
    'line item value': 8000.0,
    'pack price': 350.0,
    'unit price': 3.5,
    'weight (kilograms)': 50.0,
    'line item insurance (usd)': 10.0,
    'country_Angola': 1,  # 1 if the record belongs to Angola, else 0
    'country_Belize': 0,
    'country_Benin': 0,
    # Add all other country columns with 0 or 1
    'dosage form_Tablet': 1,  # 1 if this dosage form applies, else 0
    'dosage form_Injection': 0,
    # Add other dosage form columns with 0 or 1
    'dosage_150mg': 1,  # Add dosage columns based on the actual product
    'dosage_10mg/ml': 0,
    # Add all other dosage columns
    'manufacturing site_ABBVIE GmbH & Co.KG Wiesbaden': 1,  # Manufacturing site
    'manufacturing site_Cipla, Goa, India': 0,
    # Add all other manufacturing site columns
    'brand_Generic': 1,  # Product brand
    'brand_Truvada': 0,
    # Add all other brand columns
    'managed by_PMO - US': 1,
    'fulfill via_From RDC': 1,
    'vendor inco term_CIP': 1,
    'shipment mode_Air Charter': 1,
    'product group_ARV': 1,
    'sub classification_Adult': 1,
    'vendor_Aurobindo Pharma Limited': 1,
    'first line designation_True': 1,
    'molecule/test type_Lamivudine': 1
}
```

Random Forest Regressor Model

```
sample_df = pd.DataFrame([sample_record])

# Ensure columns match the model's training data
sample_df = sample_df.reindex(columns=X_train.columns, fill_value=0)

# Make prediction
prediction = rf_model.predict(sample_df)
print("Predicted Freight Cost (USD):", prediction[0])
```

Predicted Freight Cost (USD): 10010.757900000002

*This image is generated using AI.

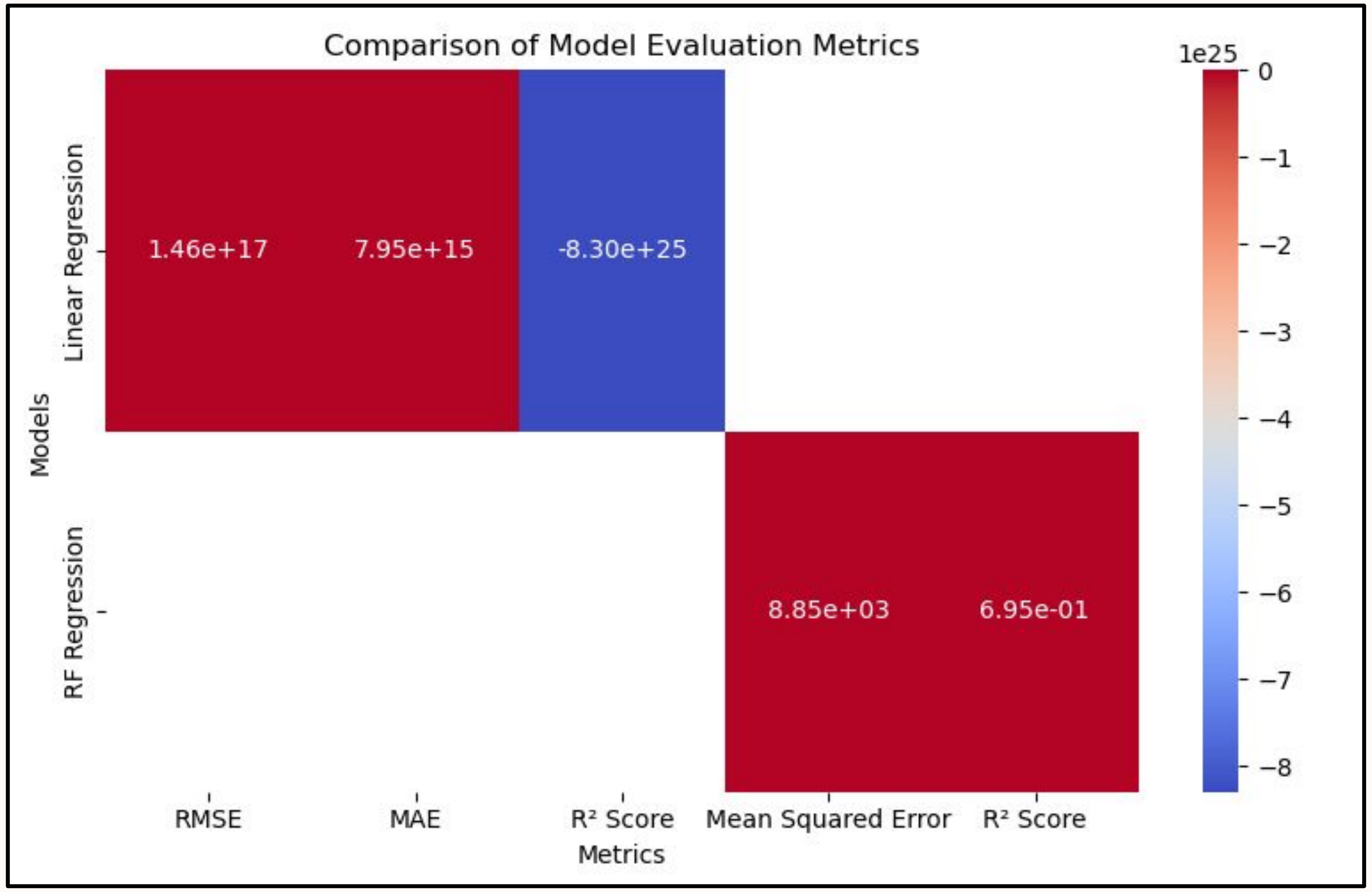
Evaluation

```
metrics_table = pd.DataFrame({
    "Metric": ["RMSE", "MAE", "R² Score"],
    "Linear Regression": [lr_rmse, lr_mae, lr_r2]
})
print("\nModel Evaluation Metrics for Linear Regression")
print(metrics_table)
```

```
Model Evaluation Metrics for Linear Regression
     Metric  Linear Regression
0      RMSE       1.459238e+17
1       MAE       7.945423e+15
2  R² Score      -8.295078e+25
```

```
metrics_table1 = pd.DataFrame({
    "Metric": ["Mean Squared Error", "R² Score"],
    "RF Regression": [mse,r2]
})
print("\nModel Evaluation Metrics for Random Forest Regression")
print(metrics_table1)
```

```
Model Evaluation Metrics for Random Forest Regression
               Metric  RF Regression
0  Mean Squared Error    8850.792686
1            R² Score       0.694837
```



- **Random Forest Regression** outperforms **Linear Regression** with a significantly lower MSE vs. extremely high RMSE and a positive R^2 score vs. a negative R^2 score
- Linear Regression shows poor fitting model.
- Random Forest Regression better handles complex, non-linear relationships, leading to a more accurate and reliable model.

Roles and Responsibilities

Sai Swetha Annem - Worked on Data Understanding ,EDA , Data Transformation and Predictions

Azmath Noorain - Worked on Linear Regression,Evaluation Metrics and Predictions

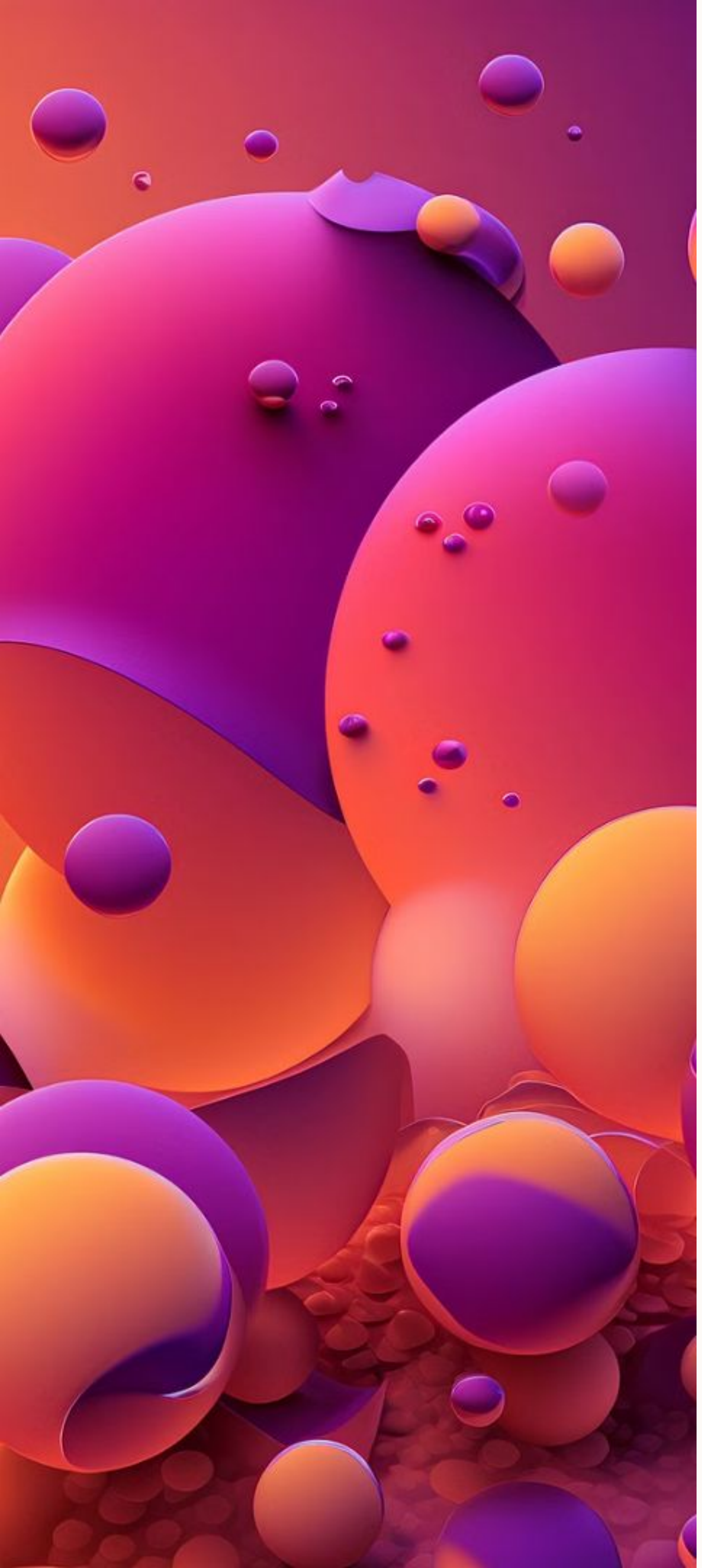
Srilakshmi Savithena - Worked on Random Forest Regressor,Evaluation Metrics and Predictions

References

- The Global Fund, https://www.theglobalfund.org/en/
- Dataset Source,

  https://catalog.data.gov/dataset/supply-chain-shipment-pricing-data-07d29
- AI ChatGPT has been used for generating images in this presentation.
- Supply Chain for HIV/AIDS,

  https://2017-2020.usaid.gov/global-health/health-areas/hiv-and-aids/technical-areas/supply-chain-hiv-and-aids-essential-health
- We have used Anaconda Juypther



Thank You

Any Questions?

*This image is generated using AI.