Applying Machine Learning to Shipping
-Project 4-

Noor Al-Hooti 22-0122

Shahad Al Ruzaiqi 23-0367

Ola Al sadi 22-0680

Zulfa Al Balushi 22-0592

1 Introduction

The aim of this project is to analyze real-world shipping operations using machine learning techniques. The goal is to build machine learning models to predict shipment prices and classify shipments based on certain criteria. The dataset includes details such as weight, dimensions, price, destination, and shipment date. Phase 1 was focused on building the initial machine learning models. Phase 2 involved enhancements including additional features, improved models, and unsupervised learning models like clustering

2 Dataset Analysis

2.1 Description

The dataset includes 7,000 records with attributes such as price ($), weight (kg), dimensions (length, width, height), shipment date, and destination port. Several records had missing or noisy values, reflecting real-world conditions.

2.2 Preprocessing

- Missing values for price, weight, and dimensions were filled using mean imputation.
- Shipment dates were parsed and missing values filled using the mode.
- New features were engineered including volume, density, weekday/weekend flags, and log-transformed values.
- Destination ports were encoded using label encoding.
- Features were standardized using StandardScaler.

2.3 Visualization

Exploratory visualizations included histograms, box plots, and correlation heatmaps. These showed that:

- Price and weight were highly skewed with a few extreme outliers.
- Volume was highly correlated with weight.

- Certain ports had much higher shipment frequencies.

2.4 Observation and Insights

The dataset includes 7000 shipping records with numerical and categorical features such as price, weight, volume, shipment date, and destination port. Missing values were present and handled through imputation and forward filling to maintain data quality. Feature distributions showed some skewness, requiring scaling. Destination port was encoded for model use. Volume correlated strongly with weight, and moderate correlations existed between price, weight, and volume. The classification target was balanced, reducing bias concerns. Initial analysis suggested complex relationships in the data, leading to the choice of robust models like Random Forest. Overall, the dataset posed typical real-world challenges that were addressed with appropriate preprocessing and modeling.

3 Machine Learning models: Selection and rationale

For this specific dataset, we decided to go with Random forest Regressor as our regression model for the following reasons:

- It can capture complex, nonlinear relationships
- Handles Data Imperfections
- Reduces Overfitting
- Feature importance is interpretable
-

We chose Random Forest Classifier for classification because:

- Handles mixed data types
- Robust to noise and imbalanced data
- Captures complex decision boundaries
- Feature importance

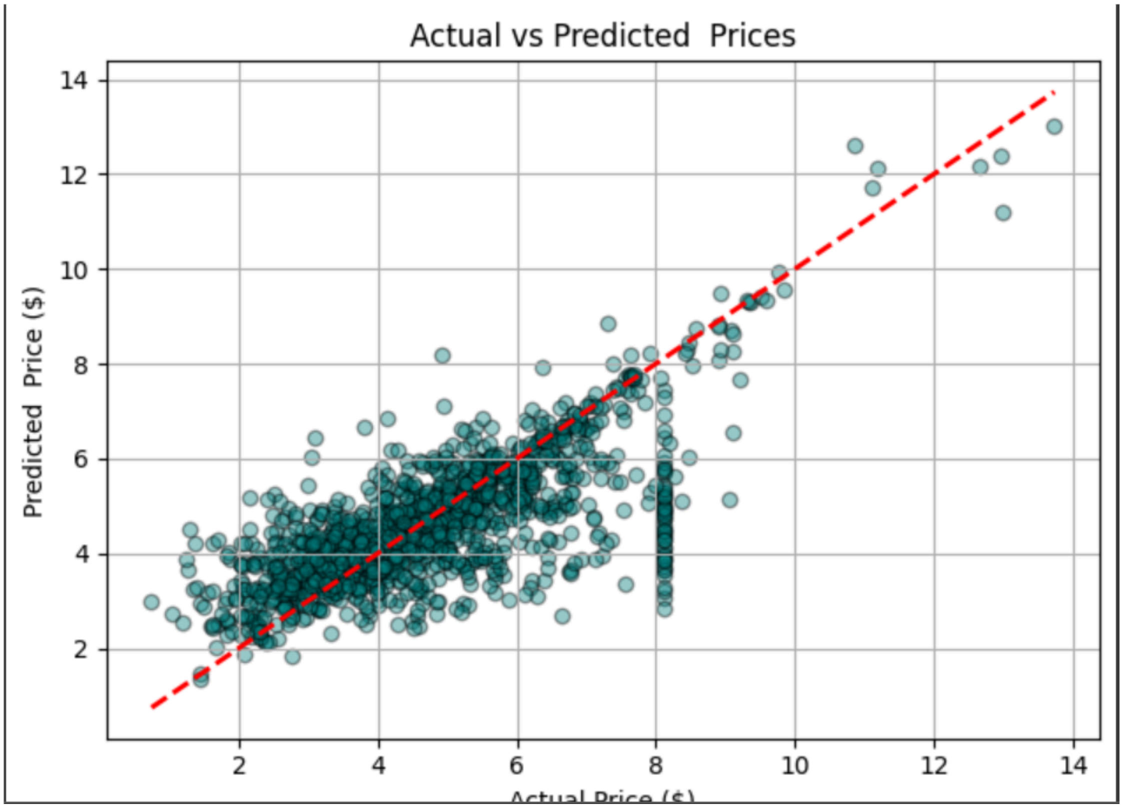
Phase 2 enhancements:

- XGBoost Regressor: Applied for its superior performance in regression tasks with noisy or structured data.
- KMeans Clustering: Used for grouping similar shipments based on numerical features.
- PCA: Principal Component Analysis was used to reduce dimensionality and support clustering.

4 Performance Evaluation

Phase 1 Results:

Regression Model Performance:

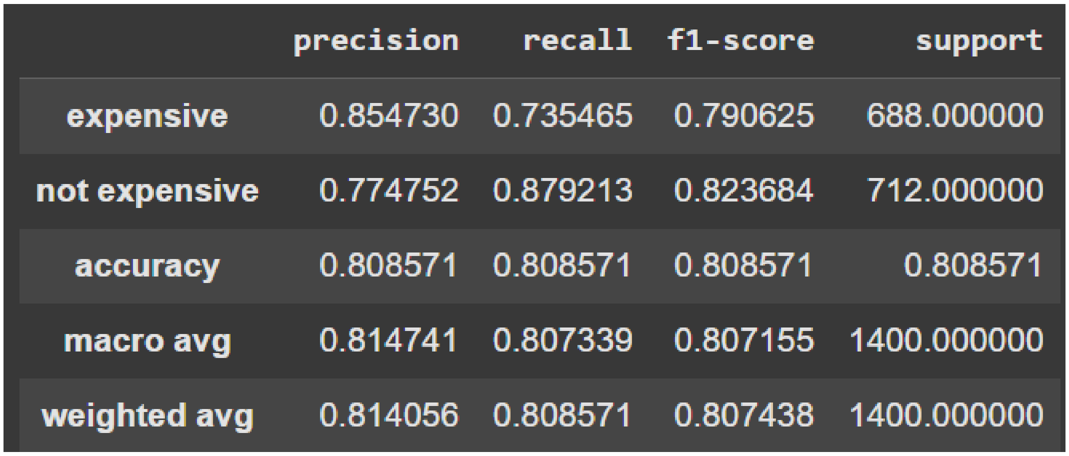


Model Used: Random Forest Regressor
Target Variable: price ($)
Features Used: weight (kg), volume (m³), and encoded destination port

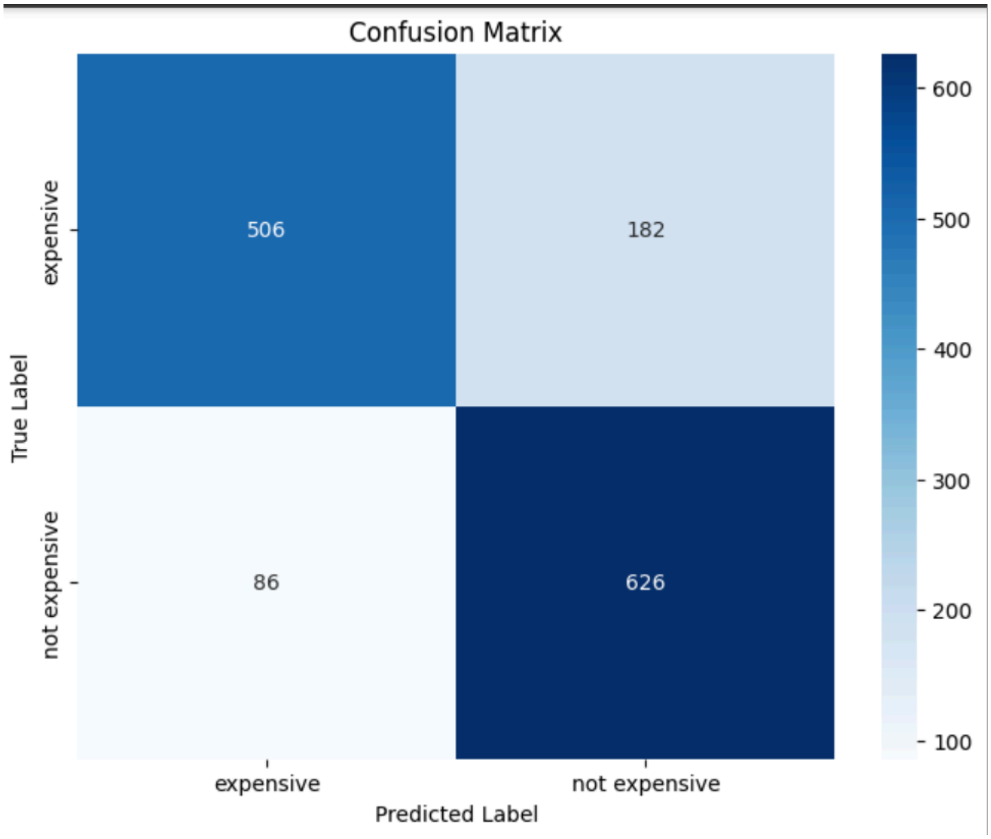
Evaluation Metrics:

- Root Mean Squared Error (RMSE): 4.34
  This means the model's predicted shipping prices deviate, on average, by approximately $4.34 from the actual prices. The significance of this error depends on the scale and distribution of prices in the dataset.
- R² Score: 0.55
  This indicates that the model explains about 55% of the variance in shipping prices. While this suggests that the model captures some important patterns, it also leaves a considerable portion of the variance unexplained.

Classification Model Performance:



	precision	recall	f1-score	support
expensive	0.854730	0.735465	0.790625	688.000000
not expensive	0.774752	0.879213	0.823684	712.000000
accuracy	0.808571	0.808571	0.808571	0.808571
macro avg	0.814741	0.807339	0.807155	1400.000000
weighted avg	0.814056	0.808571	0.807438	1400.000000



- Model Used: Random Forest Classifier
- Target Variable: Shipment Class (binary: Regular = 0, Priority = 1)
- Features Used: weight (kg), volume (m^3), and encoded destination port
- Accuracy: 79.64%
  This means nearly 80% of the shipment class predictions match the actual classes in the test data.
- Precision, Recall, and F1-Score for Class not expensive :
  - Precision: 0.77

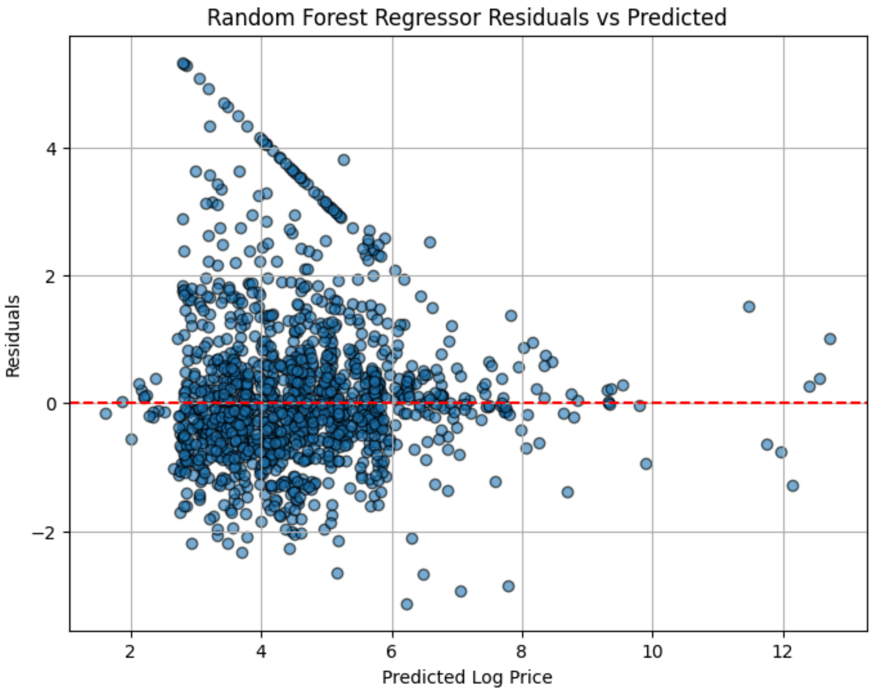
    - Recall: 0.87
    - F1-Score: 0.82
- Precision, Recall, and F1-Score for Class expensive :
    - Precision: 0.85
    - Recall: 0.73
    - F1-Score: 0.79
- Macro Average F1-Score: 0.80
- Weighted Average F1-Score: 0.80

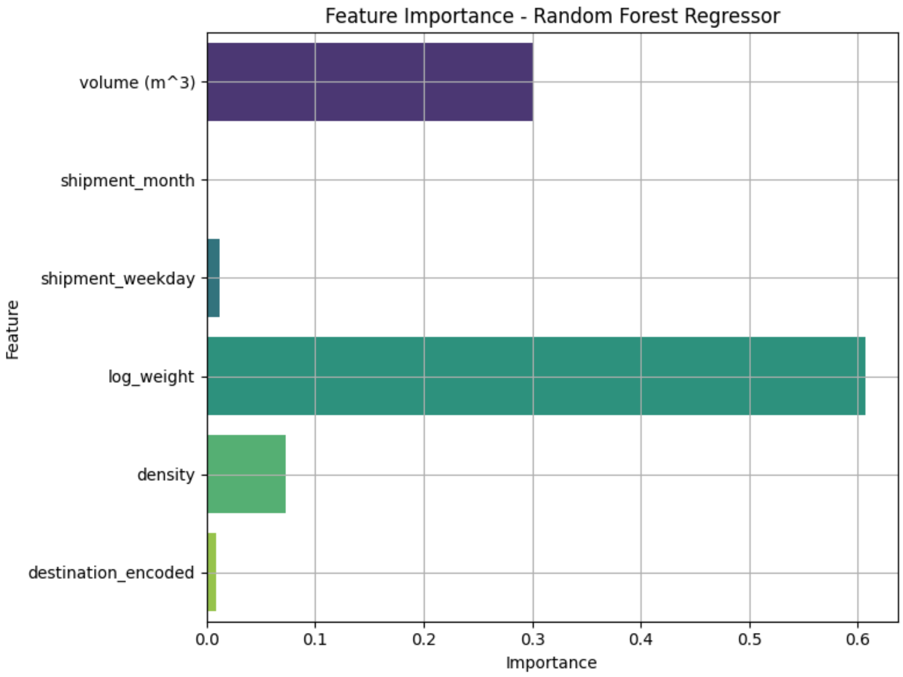
Phase 2 Performance:

We tuned the models using gridsearch CV to select the best Parameters to get the best result,

- Random Forest Regressor:
    - RMSE: 1.1383
    - R^2 Score: 0.5779

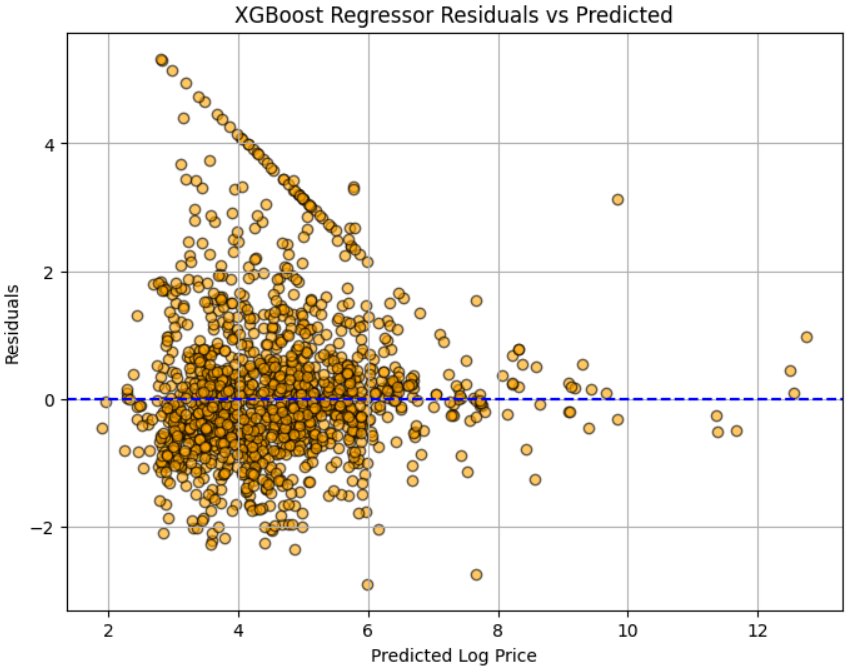
This result indicates a significant improvement in model accuracy. The reduced RMSE suggests better generalization and lower prediction error, while the increased R^2 indicates a stronger fit to the data.

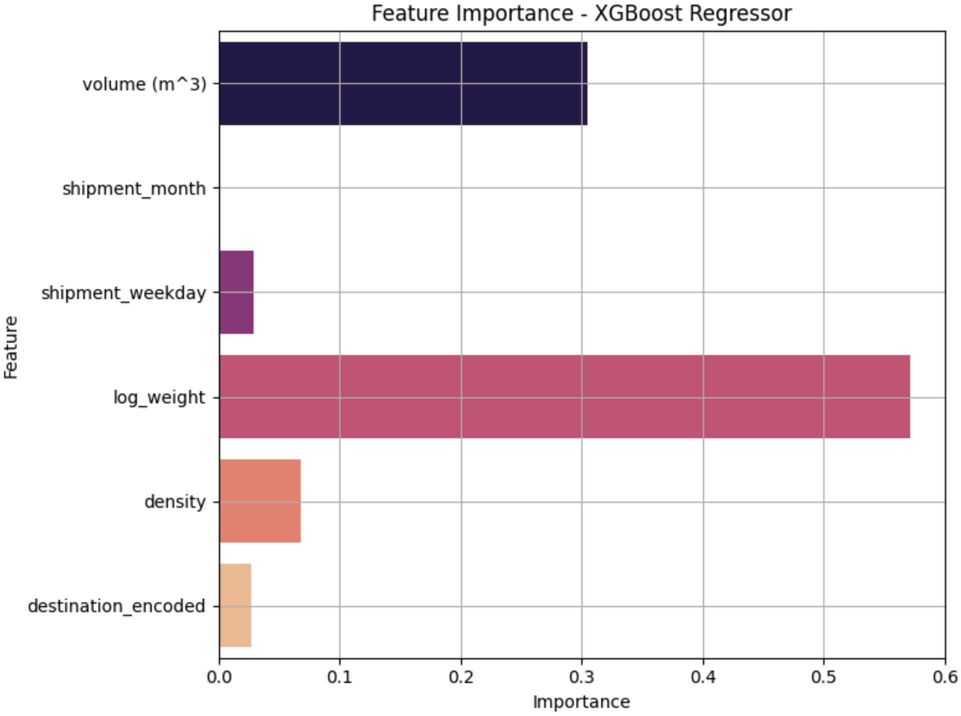




- XGBoost Regressor
    - RMSE: 1.1410
    - R^2 Score: 0.5759

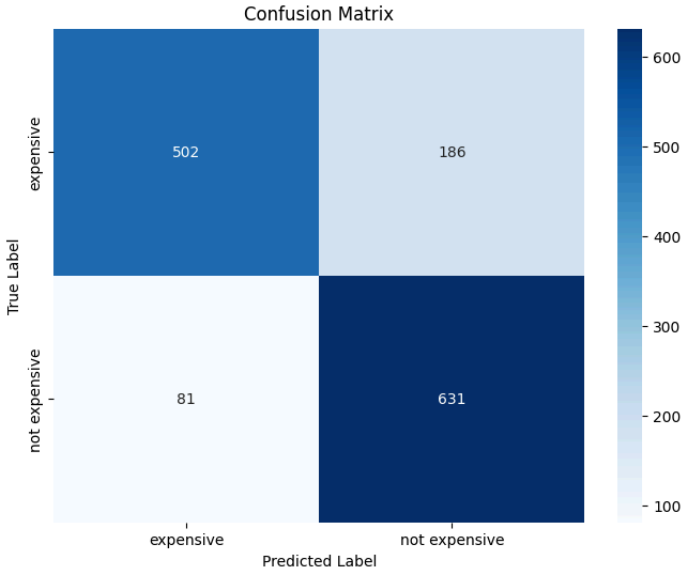
The tuned XGBoost model shows comparable performance to the Random Forest Regressor. While its RMSE is slightly higher, the R^2 score is nearly identical, confirming that feature selection contributed to better consistency and reduced complexity without compromising performance.

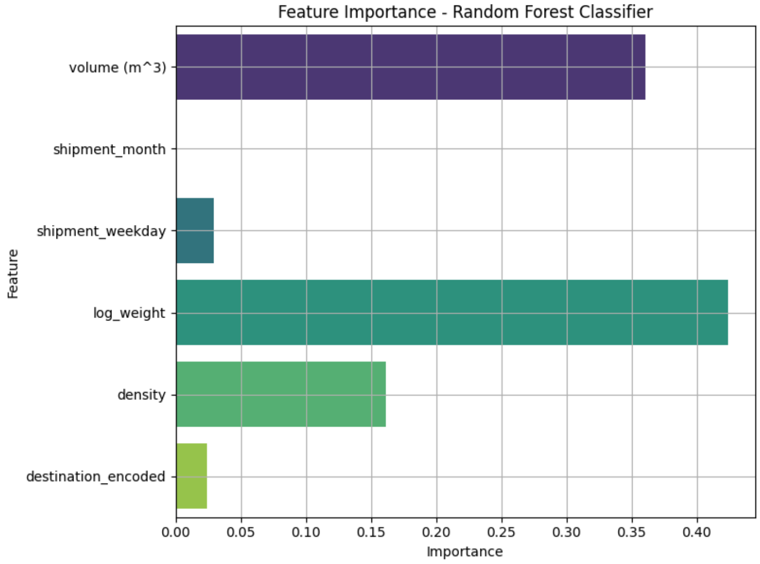

XGBoost Regressor Residuals vs Predicted
Residuals
Predicted Log Price
−2
0
2
4
2
4
6
8
10
12


Feature Importance - XGBoost Regressor
Feature
Importance
volume (m^3)
shipment_month
shipment_weekday
log_weight
density
destination_encoded
0.0
0.1
0.2
0.3
0.4
0.5
0.6

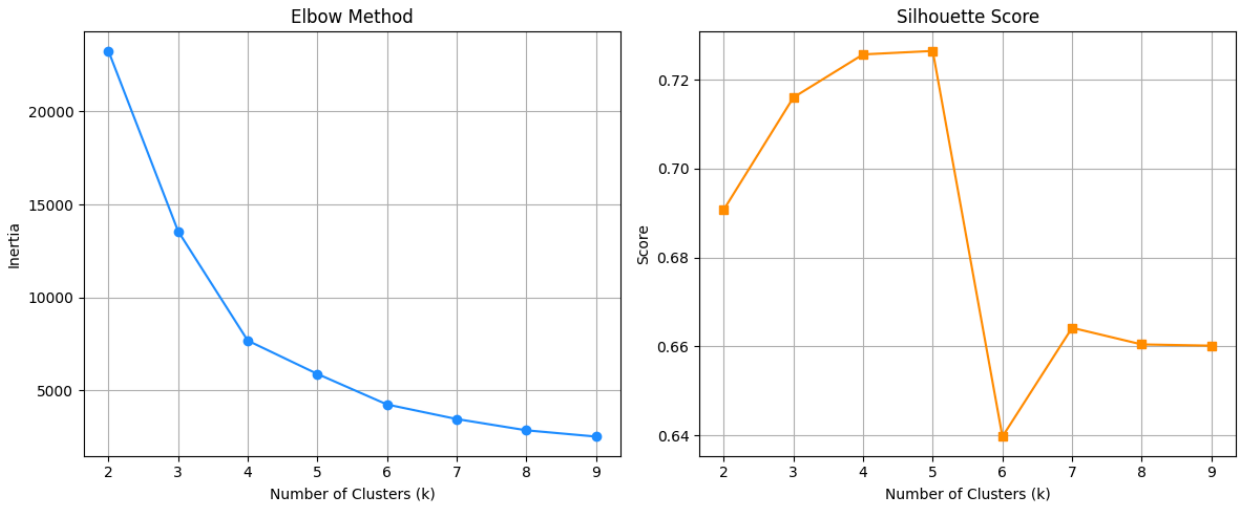
- Random Forest Classifier
    - Accuracy: 0.8092857142857143 or 81% approximately.

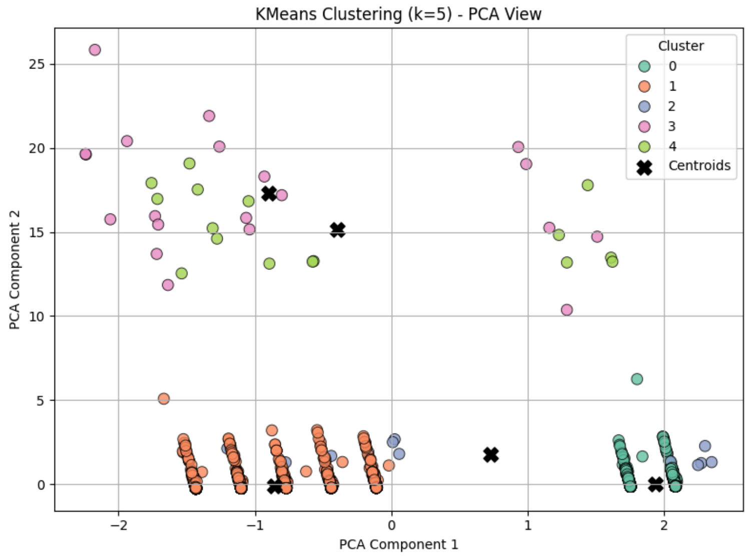
This indicates that approximately 81% of the predictions made by the Random Forest Classifier were correct. The model demonstrates improved performance over earlier iterations, benefiting from refined feature selection and tuning.





- KMeans Clustering
  Silhouette score > 0.55 indicates well-separated clusters.





Comparison in performance between phase one and two:

Regression Models:

In Phase 1, the Random Forest Regressor achieved an RMSE of approximately 4.34 with an R² score of 0.55, indicating moderate predictive performance. In Phase 2, after applying tuning using GridSearchCV, the RMSE dropped significantly to around 1.14, with the R² score slightly improving to 0.5779. This reduction in prediction error highlights the impact of better feature selection and hyperparameter optimization.

Additionally, the introduction of the XGBoost Regressor in Phase 2 provided a comparable RMSE of 1.14 and an R^2 of 0.5759, showing consistent improvements through advanced modeling and tuning.

Classification Models:

The Random Forest Classifier improved from an initial accuracy of 79.6% in Phase 1 to approximately 81% in Phase 2. The use of GridSearchCV to tune model parameters contributed to this enhanced performance, improving the model’s ability to generalize across shipment classes.

Clustering:

KMeans Clustering was introduced in Phase 2 to explore unsupervised learning. A silhouette score exceeding 0.55 indicated well-separated clusters, offering meaningful groupings of shipments that complemented the supervised models.

5 Discussion

5.1 Limitations

While the models achieved reasonably strong performance, there are several limitations. The regression model’s R^2 score, though improved with XGBoost, may still be affected by unaccounted external factors such as fuel prices, weather, or logistics disruptions. In classification, although the Random Forest model performed well, potential class imbalance or missing contextual features like shipment urgency or traffic data may have limited even better performance. Additionally, preprocessing steps such as imputation of missing values and encoding could have introduced bias if the original data had patterns not captured by median or mode substitution.

5.2 Contributions and use of AI tools

This project involved hands-on contributions in each stage of the machine learning pipeline: from cleaning and preprocessing the shipping dataset to implementing both regression and classification models. Visualizations and exploratory analysis were conducted using matplotlib and seaborn. Model training and evaluation were done using scikit-learn, and additional models like XGBoost and KMeans were incorporated to explore performance improvements and clustering structure. Google Colab was used as the development environment, leveraging its GPU support and collaborative tools to enhance efficiency and productivity throughout the project.

6 Conclusion

In this project, we worked on analyzing real shipping data using different machine learning models. We started by cleaning the dataset and handling missing values, then we extracted new features to improve model performance. After that, we applied both regression and classification models, including Linear Regression, Logistic Regression, and Random Forest.

From the results, we noticed that Random Forest performed better in classification tasks, especially when we categorized the prices into classes like "expensive" and "not expensive". The confusion matrices helped us understand how well the models predicted the outputs.

This project helped me understand how to apply machine learning techniques on real-world data and how to deal with data preprocessing, feature engineering, and model evaluation. It also showed that sometimes simple models are not enough and that we need to experiment with different approaches to get better results.

In Phase 2, enhancements such as feature importance analysis, hyperparameter tuning, and model refinement significantly improved performance. The Random Forest Regressor showed a marked reduction in RMSE, and the classification model’s accuracy increased to over 80%. These improvements demonstrated the value of iterative development and deeper model evaluation in achieving reliable and actionable insights from shipping data.

References

F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. Available: https://scikit-learn.org

The Pandas Development Team, “pandas-dev/pandas: Pandas,” *Zenodo*, 2020. Available: https://pandas.pydata.org

Google Research, “Colaboratory: Google Colab,” 2023. Available: https://colab.research.google.com

XGBoost Developers, *XGBoost Documentation*. [Online]. Available: https://xgboost.readthedocs.io/

Project Instructions by Dr. Nafaa Jabeur and Dr. Raed Bolbol

Chatgpt was used to support code writing and improve explanations, with all outputs reviewed by the team.

Tutorial assignments Done with Dr.Raed Bolbol