ESTIMATING STOCK KEEPING UNIT USING ML

Model Optimization and Tuning Phase

Objective:

To improve the performance of the selected machine learning models by tuning hyperparameters using `RandomizedSearchCV`.

Tuning Methodology:

- Used `RandomizedSearchCV` with 3-fold cross-validation
- Evaluation metric: R^2 score
- Number of iterations: 10

Tuned Model: Random Forest Regressor

Parameter Grid:

- `n_estimators`: [100, 200, 300, 400, 500]
- `max_depth`: [10, 20, 30, None]
- `min_samples_split`: [2, 5, 10]
- `min_samples_leaf`: [1, 2, 4]
- `max_features`: ['sqrt', 'log2']

Best Parameters Found:

RandomizedSearchCV identified a balanced configuration with moderate depth and optimized leaf size

Tuned Model Performance:

- R^2 Score: 0.8076
- MAE: 13.91

Observation:

- Slight performance drop compared to the untuned XGBoost model
- Indicates that XGBoost was already better suited to the dataset

Conclusion:

Although tuning improved some aspects of Random Forest, the XGBoost model remained superior

Final deployment was done using XGBoost without additional tuning

Deployment:

- Tuned model saved using `pickle`
- File Name: `sales_demand_forecasting.pkl`
- Flask web app uses this model to accept 4 recent days of input and display predicted demand

This tuning phase helped validate model robustness and confirmed the suitability of the XGBoost model for real-world deployment.