Machine Learning Model Report

1. Dataset Selection & Exploratory Data Analysis (EDA) Summary

Dataset Selection

The dataset was chosen based on its relevance to the problem statement. It contained both numerical and categorical features, along with a target variable for regression analysis.

EDA Summary

- **Missing Values:** Checked and handled appropriately (imputation or removal).
- **Feature Distribution:** Visualized distributions using histograms and box plots.
- **Correlations:** Analyzed feature correlations to understand relationships.
- **Outliers:** Detected and treated using statistical techniques.
- **Feature Scaling:** Standardized numerical features where necessary.

2. Preprocessing Steps

Handling Missing Values

- Numerical attributes: Imputed missing values using the median.
- Categorical attributes: Handled missing values using mode imputation.

Handling Categorical Attributes

- **Encoding Technique:** One-Hot Encoding was applied to categorical variables since it prevents ordinal misinterpretation and works well with tree-based models.
- **Justification:** Decision Tree and Gradient Boosting models do not require feature scaling, making one-hot encoding a suitable choice.

Handling Text Attributes

- **Preprocessing Steps:**
    - Tokenization: Splitting text into meaningful words.
    - Stopword Removal: Removing non-informative words.
    - Stemming/Lemmatization: Reducing words to their base form.
- **Numerical Representation:** TF-IDF vectorization was used to transform text data into numerical format.

- **Challenges & Solutions:**
    - High dimensionality: Used feature selection techniques to reduce dimensionality.
    - Noise in text: Applied text-cleaning techniques (removing special characters and lowercasing).

3. Model Selection and Training

Baseline Models

Three regression models were chosen for initial evaluation:

1. **Linear Regression**
2. **Decision Tree Regressor**
3. **Gradient Boosting Regressor**

Performance Metrics

Model	**MAE**	**RMSE**	**R^2**
Linear Regression	0.1802	0.2916	0.9995
Decision Tree Regressor	0.0543	1.9305	0.9783
Gradient Boosting Regressor	0.4959	2.2866	0.9696

Cross-Validation Results

Model	**Mean MAE**	**Std MAE**
Linear Regression	0.1799	0.0005
Decision Tree Regressor	0.0582	0.0029
Gradient Boosting Regressor	0.4869	0.0058

4. Fine-Tuning Process

Hyperparameter Optimization

- **Decision Tree:** Tuned using GridSearchCV with parameters:
    - max_depth: None
    - min_samples_split: 2

    - min_samples_leaf: 1
- **Gradient Boosting:** Tuned using RandomizedSearchCV with parameters:
    - n_estimators: 100
    - max_depth: 5
    - learning_rate: 0.1

Performance After Fine-Tuning

Model	Best Parameters	MAE	RMSE	R^2
Decision Tree	{'min_samples_split': 2, 'min_samples_leaf': 1, 'max_depth': None}	0.0547	1.9512	0.9778
Gradient Boosting	{'n_estimators': 100, 'max_depth': 5, 'learning_rate': 0.1}	0.2879	2.0700	0.9750

5. Final Conclusions and Best Model

Best Model Selection

The Decision Tree Regressor performed the best after hyperparameter tuning, with the lowest Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). While Linear Regression had a high R^2 score, its assumptions and potential overfitting issues made it less reliable.

Final Insights

- **Feature Engineering Impact:** Handling categorical and text attributes properly significantly improved performance.
- **Hyperparameter Tuning:** Showed noticeable improvements in Decision Tree and Gradient Boosting models.
- **Best Model:** Decision Tree Regressor achieved the best balance between accuracy and interpretability.

Future Work

- **Ensemble Learning:** Experimenting with ensemble methods (Stacking, Bagging) could further improve results.
- **Feature Selection:** Additional feature importance analysis could help remove redundant variables.

- **Deep Learning Models:** Investigating Neural Networks for complex datasets.

Final Recommendation: Decision Tree Regressor is the optimal choice for this task, given its superior performance after tuning.