Multiple Linear Regression Analysis Report

1. Introduction

Multiple Linear Regression is a statistical method used to predict the value of a **dependent variable (target)** based on two or more **independent variables (predictors)**. This report explains the steps taken to build and evaluate the model, compares different regression techniques, and provides recommendations for best performance.

---

2. Methodology

2.1 Model Equation

The model follows the equation:

Y=b0+b1X1+b2X2+…+bnXn$Y=b0+b1X1+b2X2+\ldots+bnXn$

- **Y** = Target variable (what we want to predict).
- **X_1, X_2, ..., X_n** = Predictor variables (features).
- **b_0** = Intercept (starting value when all predictors are zero).
- **b_1, b_2, ..., b_n** = Coefficients (how much each predictor affects the target).

2.2 Steps Taken

Step 1: Data Preprocessing

- **Missing Values Check:** Ensured no empty data points.
- **Exploratory Data Analysis (EDA):** Studied how features relate to each other.
- **Categorical Encoding:** Converted text categories into numbers (if needed).
- **Train-Test Split:** Divided data into:
  - **70% Training set** (to train the model).
  - **30% Testing set** (to evaluate performance).

Step 2: Model Training

- Used **LinearRegression()** from sklearn.
- Checked **coefficients** to see which features matter most.

Step 3: Performance Evaluation

Measured using:

- **R^2 Score (0-1):** How well the model explains the data (closer to 1 = better).
- **Mean Absolute Error (MAE):** Average prediction error.
- **Mean Squared Error (MSE):** Larger errors penalized more.

---

3. Model Comparison & Results

Model Type	Key Settings	R^2 Score	Notes
Standard Linear Regression	fit_intercept=True	**0.9358**	Works well when intercept is included.
	fit_intercept=False	0.7389	Much worse without intercept.
	normalize=True	**-19,155,898,675**	Normalization breaks the model.
	n_jobs=-1 (parallel)	**-21,724,334,601**	Using multiple CPUs gives bad results.
Ridge Regression (L2)	alpha=1.0	0.9357 ≈ Same	Helps prevent overfitting.
	solver='saga'	0.9353 ≈ Same	Slightly lower but stable.
Lasso Regression (L1)	max_iter=1000	0.9357 ≈ Same	Good for feature selection.
Elastic Net (L1 + L2)	l1_ratio=0.5	0.9355 ≈ Same	Balanced regularization.
Feature Scaled Regression	with_mean=True, with_std=True	0.9355 ≈ Same	Scaling doesn't help much here.

Key Findings:

1. **Standard Linear Regression** works best when fit_intercept=True (R^2 = **0.9358**).
2. **Normalization &** n_jobs **cause huge errors**—avoid them.
3. **Ridge, Lasso, and Elastic Net** perform almost the same as standard regression.
4. **Feature Scaling** doesn't improve results in this case.

4. Recommendations

Best Model Choice:

Standard Linear Regression (with fit_intercept=True**)**

- Simple and performs best.
- Avoid normalize and n_jobs—they ruin performance.

Alternative Models (If Needed):

Ridge or Lasso Regression

- Almost as good as standard regression.

- Helps prevent overfitting.

Elastic Net Regression

- Good if you want both L1 & L2 regularization.

General Tips:

✔ **Always check if intercept is needed** (fit_intercept=True is usually better).
✔ **Avoid normalization** unless necessary (it caused extreme errors here).
✔ **Feature scaling is good practice**, but it didn't help much in this case.

---

5. Conclusion

The best model is **Standard Linear Regression** with fit_intercept=True. Other models (Ridge, Lasso, Elastic Net) are also good but don't improve results significantly. Avoid normalization and multi-CPU settings (n_jobs) as they cause errors. Feature scaling is safe but not necessary here.

For future work:

- Try more datasets to confirm findings.
- Test other advanced models if needed.

Final Decision: Use **Standard Linear Regression** for best performance.