

Model Building

Three models are implemented for comparison:

1. **Linear Regression** – A simple model for predicting numerical values based on linear relationships.
 2. **Random Forest Model** – An ensemble method using decision trees for better accuracy and handling of complex patterns.
 3. **XGBoost** – An advanced gradient boosting algorithm offering high performance and efficiency.
-

Linear Regression

A baseline model that assumes a linear relationship between features and the target. Good for interpretability but may underperform on complex data.

Random Forest Model

Uses multiple decision trees and averages their predictions, improving accuracy and reducing overfitting.

XGBoost

A powerful gradient boosting framework known for its speed and accuracy. In this project:

```
python
CopyEdit
import xgboost as xgb
model_xgb = xgb.XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=5)
model = model_xgb.fit(x_train, y_train)
pred_test3 = model.predict(x_test)
```

Performance metrics like MSE, MAE, and R^2 score are calculated to evaluate the model.

Linear Regression

```
[ ] from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
```

```
[ ] from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

regressor.fit(x_train, y_train)

pred_test=regressor.predict(x_test)
print("test_mse: ", mean_squared_error(y_test, pred_test))
print("test_mae: ", mean_absolute_error(y_test, pred_test))
print("R2_score: ", r2_score(y_test, pred_test))
```

```
test_mse: 0.879100826664016
test_mae: 0.1957826268679382
R2_score: -27.99985070092153
```

Random Forest Model

```
[ ] from sklearn.ensemble import RandomForestRegressor
model_rf= RandomForestRegressor(n_estimators= 200 , max_depth=5)
model2= model_rf.fit(x_train, y_train)
```

```
pred_test2= model2.predict(x_test)
print("test_mse: ", mean_squared_error(y_test, pred_test2))
print("test_mae: ", mean_absolute_error(y_test, pred_test2))
print("R2_score: ", r2_score(y_test, pred_test2))
```

```
test_mse: 0.020629835668726747
test_mae: 0.0990796086456456
R2_score: 0.31946127653196765
```

Xgboost

```
[ ] import xgboost as xgb
model_xgb = xgb.XGBRegressor(n_estimators=100, learning_rate=0.1, max_depth=5)
model= model_xgb.fit(x_train, y_train)
```

```
[ ] pred_test3= model.predict(x_test)
print("test_mse: ", mean_squared_error(y_test, pred_test3))
print("test_mae: ", mean_absolute_error(y_test, pred_test3))
print("R2_score: ", r2_score(y_test, pred_test3))
```

```
test_mse: 0.02021150707977389
test_mae: 0.09780533089943558
R2_score: 0.3332611345864712
```