

## Model Optimization and Tuning Phase

Date	28 June 2025
Team ID	SWTID1749705685
Project Title	Movie Box Office Gross Prediction using Machine Learning
Maximum Marks	10 Marks

### Model Optimization and Tuning Phase

The Model Optimization and Tuning Phase focuses on improving model performance by adjusting hyperparameters, evaluating different algorithms, and selecting the most suitable model. This ensures accurate and efficient predictions of movie box office revenue.

### Hyperparameter Tuning Documentation (6 Marks):

Model	Tuned Hyperparameters	Optimal Values
Linear Regression	<pre>lr = LinearRegression() lr_params = {     'fit_intercept': [True, False],     'positive': [True, False] } grid_lr = GridSearchCV(     lr,     lr_params,     cv=5,     scoring='r2',     n_jobs=-1 ) grid_lr.fit(x_train, y_train)</pre>	<pre>print("Optimal Hyperparameters:", grid_lr.best_params_) print("R2 Score on Test Set:", grid_lr.score(x_test, y_test))</pre> <p>✓ 5.7s</p> <p>Optimal Hyperparameters: {'fit_intercept': True, 'positive': True} R2 Score on Test Set: 0.6676333297180277</p>
XGBoost Regressor	<pre>xgb = XGBRegressor(random_state=42, objective='reg:squarederror') xgb_params = {     'n_estimators': [100, 200],     'max_depth': [3, 6, 10],     'learning_rate': [0.01, 0.1],     'subsample': [0.8, 1.0] } grid_xgb = GridSearchCV(     xgb,     xgb_params,     cv=5,     scoring='r2',     n_jobs=-1 ) grid_xgb.fit(x_train, y_train)</pre>	<pre>print("Optimal Hyperparameters:", grid_xgb.best_params_) print("R2 Score on Test Set:", grid_xgb.score(x_test, y_test))</pre> <p>✓ 10.0s</p> <p>Optimal Hyperparameters: {'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100, 'subsample': 0.8} R2 Score on Test Set: 0.7040000265300657</p>

Random Forest	<pre>rf = RandomForestRegressor(random_state=42) rf_params = {     'n_estimators': [100, 200],     'max_depth': [None, 10, 20],     'min_samples_split': [2, 5],     'min_samples_leaf': [1, 2] } grid_rf = GridSearchCV(     rf,     rf_params,     cv=5,     scoring='r2',     n_jobs=-1 ) grid_rf.fit(x_train, y_train)</pre>	<pre>grid_rf.fit(x_train, y_train)  print("Optimal Hyperparameters:", grid_rf.best_params_) print("R2 Score on Test Set:", grid_rf.score(x_test, y_test))</pre> <p>✓ 400s</p> <p>Optimal Hyperparameters: {'max_depth': 10, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 200} R2 Score on Test Set: 0.7002211740410094</p>
---------------	--	--

### Performance Metrics Comparison Report (2 Marks):

Model	Baseline Metric	Optimized Metric
Linear Regression	0.4871	0.6676
XGBoost Regressor	0.5425	0.7087
Random Forest	0.5586	0.6988

### Final Model Selection Justification (2 Marks):

Final Model	Reasoning
Linear Regression	Linear Regression was chosen for its <b>simplicity, interpretability</b> , and relatively <b>high R<sup>2</sup> score (0.6679)</b> . Despite other models performing better, Linear Regression met the project's goal of building a transparent and explainable model suitable for academic and deployment purposes.