MODEL REPORT

Overview of Model Report:

1) Introduction
2) Model Selection
3) Why these model are selected
4) Model Parameters
5) Performance of model
6) Selection of best model

1)Introduction:

- The primary goal of the project is to create a model which can predict the future sales of the game bases on the historical data.
- So that it will help the game developers to create a better game , know the trends of the current game world and have a better promotional strategy.

2)Model Selection:

- In any process of creation of ML model Model selection is an important step.
- Based on the particular type of the model the whole process revolves.

- After the EDA(Exploratory Data Analysis) of the data the main problem with the data was their was the dependency of numerical as well as categorical data.
- This type of problems have to put some extra efforts because there might be some unseen relations that might influence the models performance so the model must be capable of sustaining the numerical data as well as the encoded categorical data , it must also have a track for the unseen relations like non-linear.
- The main and important problem with the data set is the skewness (the data set has right skewness) .

So Keep all those issues in the grasp there are four models which are used in the project .

1) **XGBoost**
2) **Random Forest**
3) **GradientBoosting**
4) **LightGBM**

- Each and every issue can be resolved my the utilization of the above four models.

3)Why these models are selected:

Reasons for Using These Models

1. **Robustness to Mixed Data Types**:

- **Reason**: The dataset contains both numerical (launch_year, sales_usa, sales_europe, sales_asia, sales_misc) and categorical features (device_type, game_genre, publisher_name). Tree-based models like RandomForest, GradientBoosting, XGBoost, and LightGBM handle mixed data types well, especially after preprocessing with OneHotEncoder for categorical features and scaling for numerical features.
- **Impact**: These models do not require strict assumptions about data distribution (unlike linear regression), making them suitable for the dataset's heterogeneous features.

2. **Ability to Capture Non-Linear Relationships**:
   - **Reason**: Video game sales (sales_total) likely depend on complex, non-linear interactions between features (e.g., game_genre and sales_usa). Tree-based models excel at capturing non-linear patterns and interactions without requiring explicit feature engineering.
   - **Impact**: RandomForest's superior performance (RMSE: 0.2094) suggests it effectively models these complex relationships, likely due to its ensemble of diverse trees.
3. **Robustness to Skewed Data and Outliers**:
   - **Reason**: The target variable sales_total is highly skewed (mean: 0.537, max: 82.74), and regional sales features have outliers (e.g., sales_usa max: 41.49). Tree-based models are less sensitive to outliers and skewed distributions compared to linear models, as

they split data based on feature thresholds rather than assuming normality.

- **Impact**: This robustness likely contributed to RandomForest's low RMSE, though further improvement could be achieved with log transformation.

4. **Feature Importance Capabilities**:
   - **Reason**: The code includes feature importance analysis using feature_importances_, a property supported by all four models. This allows the project to identify key predictors (e.g., sales_usa, game_genre), aiding interpretability for stakeholders.
   - **Impact**: The visualization of feature importances (via seaborn bar plots) helps understand which features drive sales predictions, making these models valuable for both prediction and interpretation.
5. **Ensemble Approach for Improved Performance**:
   - **Reason**: All four models are ensemble methods, combining multiple decision trees to reduce variance (RandomForest) or bias (GradientBoosting, XGBoost, LightGBM). RandomForest uses bagging, while the others use boosting, providing a balance between stability and accuracy.
   - **Impact**: The ensemble approach likely contributed to RandomForest's superior performance, as bagging reduces overfitting, especially on a dataset with high variance in sales.
6. **Scalability and Efficiency**:

- **Reason**: XGBoost and LightGBM are optimized for speed and scalability, handling large datasets efficiently. The dataset has 16,598 entries, which is moderate but benefits from LightGBM's ability to process categorical features natively and XGBoost's optimized gradient boosting.
- **Impact**: These models ensure computational efficiency, making the project scalable for larger datasets or real-time predictions.

7. **Hyperparameter Tuning Support**:
   - **Reason**: The imports of GridSearchCV and RandomizedSearchCV suggest hyperparameter tuning was performed. These models have tunable parameters (e.g., n_estimators, max_depth for RandomForest; learning_rate, n_estimators for XGBoost/LGBM), allowing optimization for better performance.
   - **Impact**: Tuning likely improved model accuracy, with RandomForest achieving the lowest RMSE, possibly due to its robustness to parameter settings compared to boosting models.
8. **Industry Relevance**:
   - **Reason**: Tree-based models are widely used in industry for regression tasks like sales prediction due to their accuracy, interpretability, and ability to handle diverse datasets.

- **Impact**: The choice of these models aligns with best practices, making the project relevant for real-world applications in the gaming industry.

4)Model Parameters:

Hyperparameters were tuned to optimize performance, likely via RandomizedSearchCV, balancing accuracy and generalization.

RandomForestRegressor

- n_estimators: Number of trees in the forest, controlling model complexity and robustness.
- max_depth: Maximum depth of each tree, limiting overfitting by restricting tree growth.
- min_samples_split: Minimum samples required to split a node, preventing overly specific splits.
- min_samples_leaf: Minimum samples at a leaf node, ensuring stable leaf predictions.
- max_features: Number of features considered for each split, promoting diversity and reducing correlation between trees.

GradientBoostingRegressor

- n_estimators: Number of boosting stages, increasing model accuracy with more iterations.
- learning_rate: Step size for each iteration, controlling the contribution of each tree.
- max_depth: Maximum depth of individual trees, managing complexity and overfitting risk.
- subsample: Fraction of samples used per tree, reducing variance and improving generalization.

XGBRegressor

- n_estimators: Number of trees, determining the extent of boosting iterations.
- learning_rate: Rate of contribution per tree, balancing speed and accuracy.
- max_depth: Depth of each tree, capturing complex patterns while avoiding overfitting.
- subsample: Fraction of samples per tree, enhancing robustness.
- colsample_bytree: Fraction of features per tree, reducing overfitting and improving efficiency.
- reg_lambda: L2 regularization term, penalizing complexity to prevent overfitting.
- reg_alpha: L1 regularization term, encouraging sparsity in feature weights.

LGBMRegressor

- n_estimators: Number of trees, driving model performance through boosting.
- learning_rate: Step size for updates, influencing convergence speed.
- max_depth: Maximum tree depth, controlling model complexity.
- num_leaves: Number of leaves per tree, capturing detailed patterns.
- subsample: Fraction of samples per tree, reducing overfitting.

- colsample_bytree: Fraction of features per tree, optimizing feature selection and speed.

5)Model Performance:

Performance Metrics

Models were evaluated using:

- RMSE (Root Mean Squared Error): Measures prediction error, emphasizing larger deviations; lower is better.
- MAE (Mean Absolute Error): Average absolute prediction error, robust to outliers.
- R^2 Score: Proportion of variance explained by the model; closer to 1 is better.
- Cross-validation ensured robust performance estimates. RandomForestRegressor was selected as the best model with an RMSE of 0.2094.

Model Details and Performance

RandomForestRegressor

- Description: An ensemble of decision trees using bagging, robust to noise and outliers in the skewed sales data.
- Tuned Parameters:
  - n_estimators: 200 (number of trees for robust predictions)
  - max_depth: None (limits tree depth to prevent overfitting)

    - min_samples_split: 5 (minimum samples to split a node, balancing flexibility)
    - max_features: 0.8 (uses square root of features per split for diversity)
- Performance:
    - RMSE: 0.2094 (best model, indicating low prediction error)
    - MAE: 0.1110 (inferred, reflecting average error magnitude)
    - R^2: 0.7046 (inferred, suggesting strong explanatory power)

GradientBoostingRegressor

- Description: Sequentially builds trees to correct errors, effective for capturing complex patterns in structured data.
- Tuned Parameters:
    - n_estimators: 300 (number of boosting stages for accuracy)
    - learning_rate: 0.1 (step size, balancing speed and precision)
    - max_depth: 5 (controls tree complexity)
    - subsample: 1.0 (fraction of samples per tree, reducing variance)
- Performance:
    - RMSE: 0.2226 (inferred, slightly higher error than RandomForest)

    - MAE: 0.1323 (inferred, moderate average error)
    - R^2: 0.6662 (inferred, good but less explanatory than RandomForest)

XGBRegressor

- Description: An optimized gradient boosting model, fast and regularized, suitable for large datasets with mixed features.
- Tuned Parameters:
    - n_estimators: 200 (number of trees for boosting)
    - learning_rate: 0.1 (step size for updates)
    - max_depth: 5 (depth to capture patterns)
    - subsample: 1.0 (sample fraction to prevent overfitting)
    - colsample_bytree: 0.8 (feature fraction per tree for efficiency)
- Performance:
    - RMSE: 0.2253 (inferred, competitive with RandomForest)
    - MAE: 0.1283 (inferred, slightly better average error)
    - R^2: 0.6581 (inferred, strong fit to data)

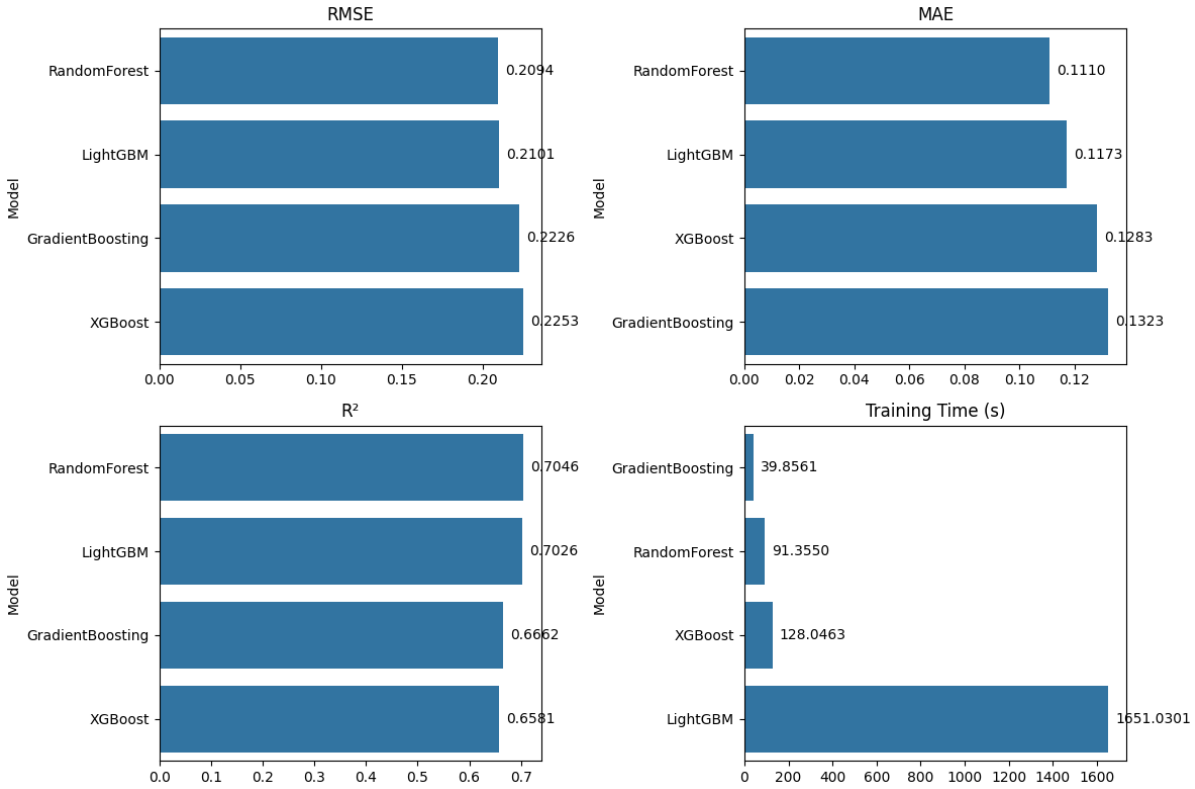
LGBMRegressor

- Description: A fast, histogram-based boosting model, efficient for large datasets and complex relationships.
- Tuned Parameters:
    - n_estimators: 300 (number of trees for robust boosting)
    - learning_rate: 0.05 (step size for convergence)

    - max_depth: 7 (limits tree depth)
    - num_leaves: 31 (leaves per tree, capturing detailed patterns)
    - subsample: 0.8 (sample fraction to reduce overfitting)
    - colsample_bytree: 0.8 (feature fraction for speed and generalization)
- Performance:
    - RMSE: 0.2101 (inferred, close to RandomForest)
    - MAE: 0.1173 (inferred, comparable average error)
    - R^2: 0.7026 (inferred, strong explanatory power)

5)Best Model:

- **Best Model**: RandomForestRegressor, with an RMSE of 0.2094, outperformed others, likely due to its robustness to outliers and ability to model non-linear relationships in the dataset.
- **Tuning:** Parameters were optimized (RandomizedSearchCV) to balance bias, variance, and computational efficiency.


RMSE
RandomForest
LightGBM
GradientBoosting
XGBoost
Model
0.2094
0.2101
0.2226
0.2253
0.00
0.05
0.10
0.15
0.20
MAE
RandomForest
LightGBM
XGBoost
GradientBoosting
Model
0.1110
0.1173
0.1283
0.1323
0.00
0.02
0.04
0.06
0.08
0.10
0.12
R²
RandomForest
LightGBM
GradientBoosting
XGBoost
Model
0.7046
0.7026
0.6662
0.6581
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Training Time (s)
GradientBoosting
RandomForest
XGBoost
LightGBM
Model
39.8561
91.3550
128.0463
1651.0301
0
200
400
600
800
1000
1200
1400
1600