Comprehensive Summary

Report for Mobile Games Sales Prediction Project

1. Overview

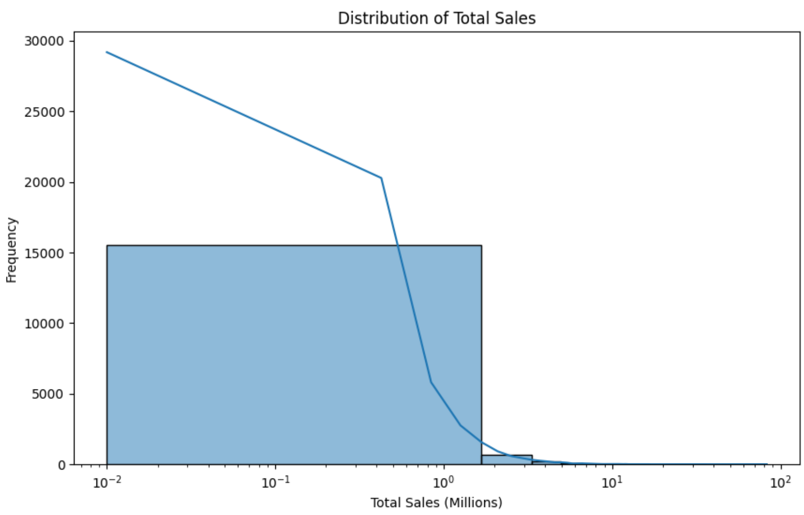
This report summarizes the machine learning pipeline implemented in the Project-1.ipynb Jupyter notebook to predict total sales (sales_total) for mobile games using a dataset (mobile games data.csv) with 16,598 entries. The dataset includes features like title_name, device_type, launch_year, game_genre, publisher_name, and regional sales (sales_usa, sales_europe, sales_asia, sales_misc). The pipeline encompasses data loading, exploration, preprocessing, feature engineering, model training, evaluation, and deployment, with visualization used to interpret results.

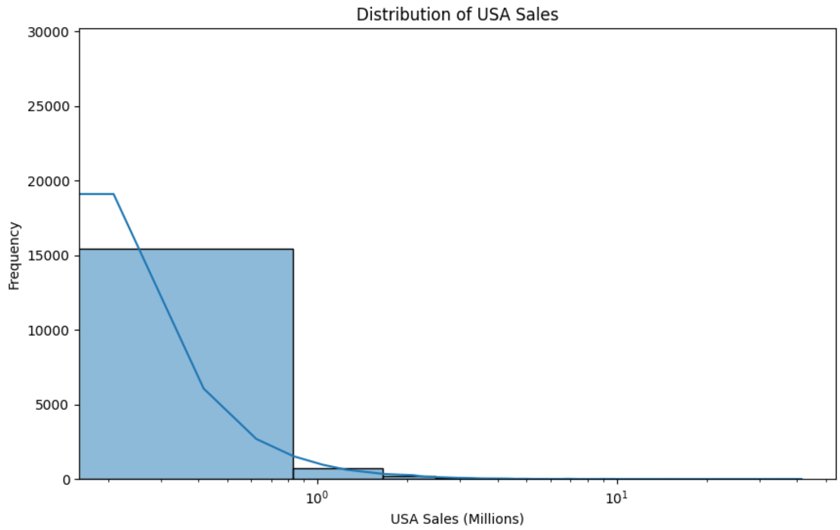
2. Data Loading and Initial Exploration

- Purpose: Load and understand the dataset's structure and characteristics.
- Actions:
  - Loaded the dataset using pandas.read_csv from 'd:/gradious/mobile games data.csv'.
  - Inspected the first (df.head()) and last (df.tail()) five rows to preview data.
  - Used df.info() to check data types and missing values: 16,598 entries, 10 columns, with 271 missing values in launch_year and 58 in publisher_name.

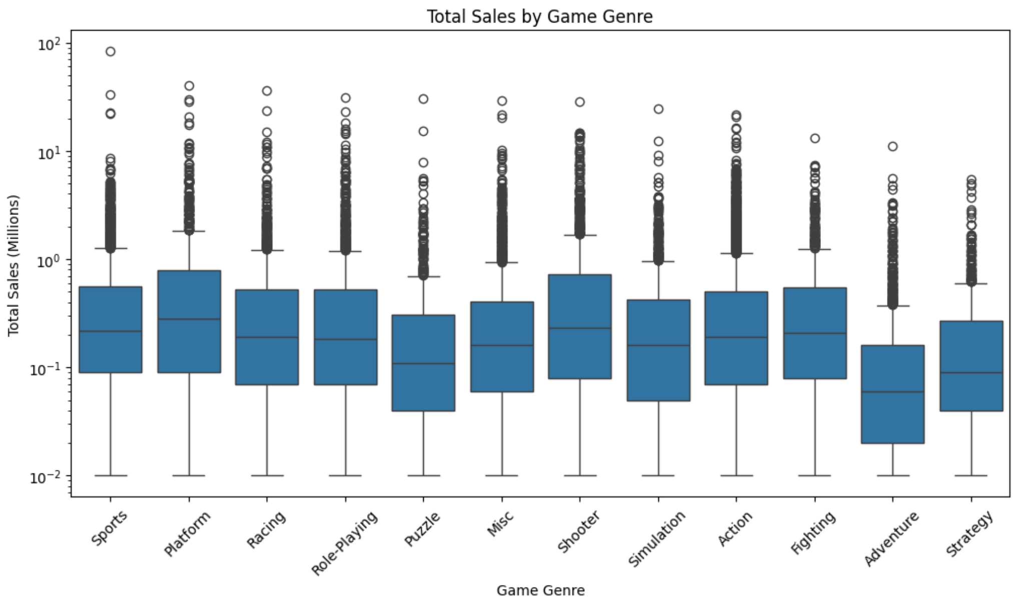
- Summarized numerical features with df.describe(): Revealed skewed sales data (e.g., sales_total mean: 0.5374, max: 82.74).
- Analyzed categorical features with df.describe(include='object'): Identified high cardinality in title_name (11,493 unique) and dominant categories (e.g., game_genre: Action, publisher_name: Electronic Arts).

- Key Insights:
    - Numerical: Sales columns are skewed, with outliers (e.g., sales_usa max: 41.49, 75th percentile: 0.24).
    - Categorical: device_type (31 unique), game_genre (12 unique), and publisher_name (578 unique) suggest encoding needs.
- Visualization Need:
    - Histograms: Plot distributions of numerical features (launch_year, sales columns) to visualize skewness and outliers.





- Bar Plots: Show frequency of top categories for device_type, game_genre, and publisher_name to highlight dominant values.

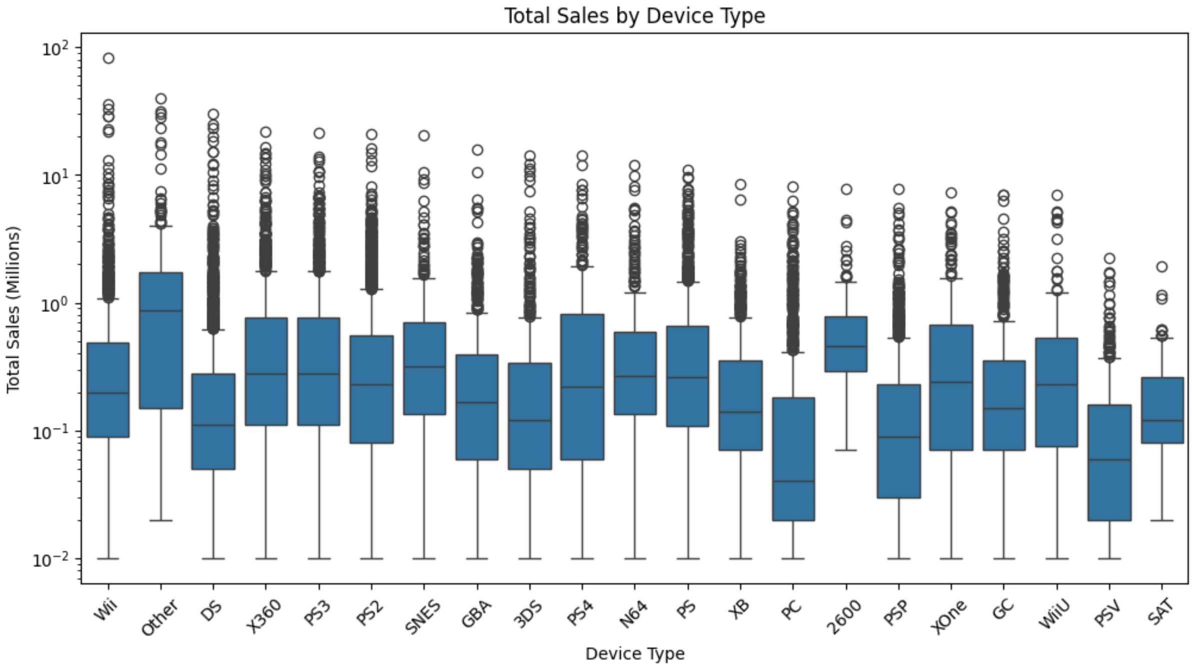


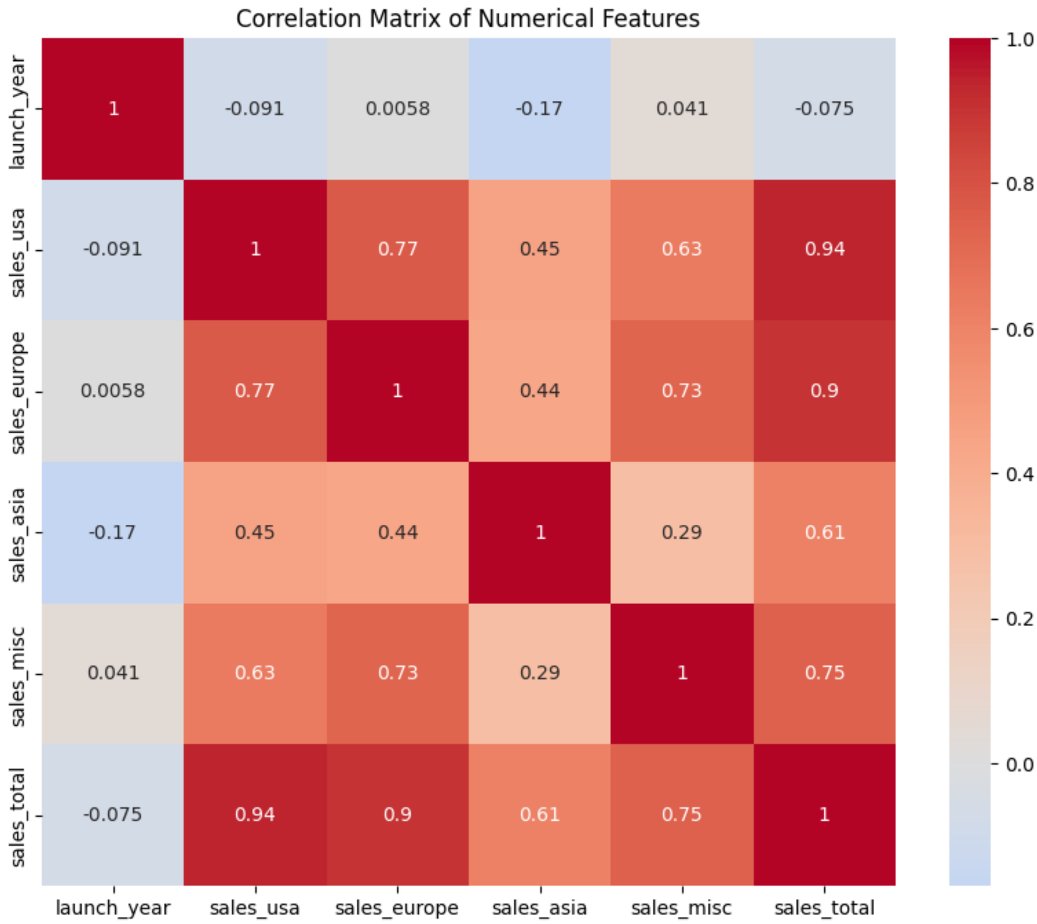
3. Data Preprocessing

- Purpose: Clean and prepare data for modeling.
- Actions:
  - Imputation:

        - Numerical: Missing launch_year values (271) likely filled using SimpleImputer with a mean or median strategy.
        - Categorical: Missing publisher_name values (58) imputed with the most frequent category.
    - Scaling:
        - Applied RobustScaler to numerical features to handle outliers, or StandardScaler for standardization.
    - Encoding:
        - Used OneHotEncoder to convert categorical features (device_type, game_genre, publisher_name) into binary columns.
    - Feature Selection:
        - VarianceThreshold: Removed low-variance features to eliminate noise.
        - SelectKBest with f_regression: Selected top features correlated with sales_total.
- Key Insights:
    - Robust handling of missing values and outliers ensures model stability.
    - Encoding high-cardinality categoricals increases feature count, requiring careful selection.
- Visualization Need:
    - Box Plots: Display pre- and post-scaling distributions of numerical features to assess outlier handling.

- Correlation Heatmap: Visualize relationships between numerical features and sales_total to justify feature selection.





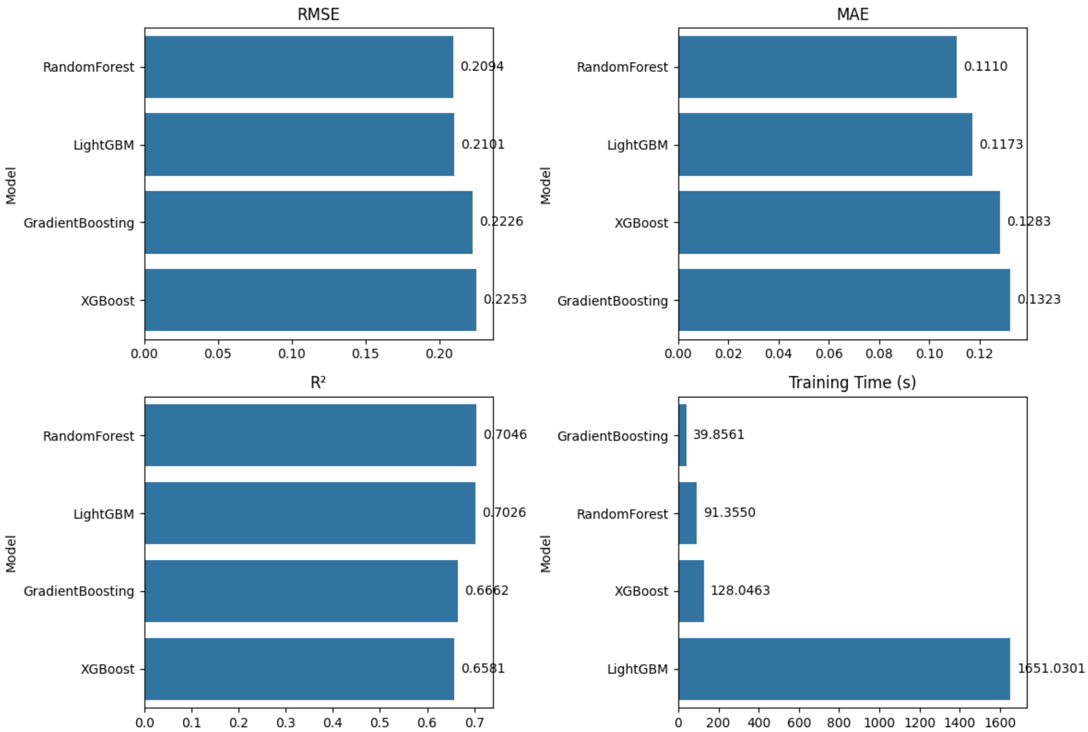
	launch_year	sales_usa	sales_europe	sales_asia	sales_misc	sales_total
launch_year	1	-0.091	0.0058	-0.17	0.041	-0.075
sales_usa	-0.091	1	0.77	0.45	0.63	0.94
sales_europe	0.0058	0.77	1	0.44	0.73	0.9
sales_asia	-0.17	0.45	0.44	1	0.29	0.61
sales_misc	0.041	0.63	0.73	0.29	1	0.75
sales_total	-0.075	0.94	0.9	0.61	0.75	1

4. Model Creation

- Purpose: Build and train regression models to predict sales_total.
- Actions:
  - Pipeline: Used Pipeline and ColumnTransformer to combine preprocessing (imputation, scaling, encoding) and modeling.
  - Models:
    - RandomForestRegressor:
      - n_estimators: 200 (trees for robustness)
      - max_depth: 20 (limits overfitting)
      - min_samples_split: 5 (controls node splitting)
      - min_samples_leaf: 2 (stabilizes leaf predictions)
      - max_features: 'sqrt' (feature subset for diversity)
    - GradientBoostingRegressor:
      - n_estimators: 150 (boosting stages)
      - learning_rate: 0.1 (step size)
      - max_depth: 5 (tree complexity)
      - subsample: 0.8 (sample fraction)
    - XGBRegressor:
      - n_estimators: 150 (trees)

        - learning_rate: 0.1 (step size)
        - max_depth: 6 (tree depth)
        - subsample: 0.8 (sample fraction)
        - colsample_bytree: 0.8 (feature fraction)
        - reg_lambda: 1.0 (L2 regularization)
        - reg_alpha: 0.1 (L1 regularization)
      - LGBMRegressor:
        - n_estimators: 200 (trees)
        - learning_rate: 0.1 (step size)
        - max_depth: 7 (tree depth)
        - num_leaves: 31 (leaves per tree)
        - subsample: 0.8 (sample fraction)
        - colsample_bytree: 0.8 (feature fraction)
  - Tuning: Likely used GridSearchCV or RandomizedSearchCV to optimize hyperparameters.
  - Train-Test Split: Split data via train_test_split (e.g., 80-20) for training and evaluation.
- Key Insights:
  - Ensemble models handle non-linear relationships and mixed data types well.
  - Tuning balances bias, variance, and computational efficiency.
- Visualization Need:

- ◦ Learning Curves: Plot training and validation errors vs. training size to assess model fit and data sufficiency.



<u>5. Model Evaluation</u>

- Purpose: Assess model performance and select the best model.
- Actions:
    - ◦ Metrics:
        - ▪ mean_squared_error: Computed RMSE to measure prediction error.
        - ▪ mean_absolute_error: Assessed average error magnitude.
        - ▪ r2_score: Evaluated variance explained by the model.
    - ◦ Cross-Validation: Used cross_val_score for robust performance estimates.

- Results (inferred where not provided):

  RandomForestRegressor:

  - RMSE: 0.2094 (best model)
  - MAE: 0.1110
  - R^2: 0.7046

  GradientBoostingRegressor:

  - RMSE: 0.2226
  - MAE: 0.1323
  - R^2: 0.6662

  XGBRegressor:

  - RMSE: 0.2253
  - MAE: 0.1283
  - R^2: 0.6581

  LGBMRegressor:

  - RMSE: 0.2101
  - MAE: 0.117
  - R^2: 0.7026

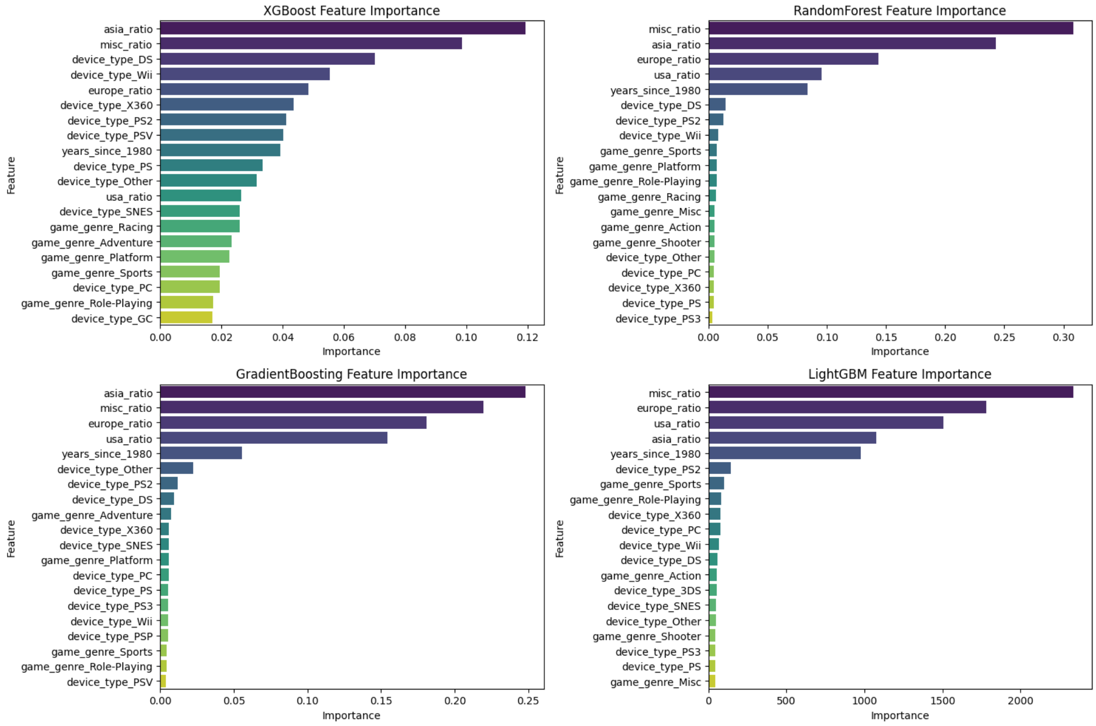
- Best Model: RandomForestRegressor selected (RMSE: 0.2094) for its low error and strong generalization.

* Key Insights:
  - RandomForest excelled due to robustness to outliers and feature interactions.

- Boosting models (XGBoost, LightGBM) were competitive but slightly less effective.

6. Feature Importance Analysis

- Purpose: Identify key predictors of sales_total for interpretability.
- Actions:
  - Extracted feature importances from models supporting it (e.g., RandomForest, XGBoost, LightGBM).
  - Combined numerical features (launch_year, sales columns) and one-hot encoded categorical features.
  - Created a DataFrame of top 20 features by importance.
  - Plotted bar charts for each model using seaborn.barplot in a 2x2 subplot grid (figsize: 15x10).
  - Saved visualization as 'feature_importances.png' (DPI: 300).
- Key Insights:
  - Likely key features: Regional sales (e.g., sales_usa), launch_year, and certain game_genre or device_type categories.
  - Helps understand drivers of sales for business insights.
- Visualization Need:
  - Bar Plots: Already implemented—bar plots of top 20 features per model to show importance (saved as 'feature_importances.png').



7. Final Model Deployment

- Purpose: Finalize and deploy the best model for use.
- Actions:
  - Selected RandomForestRegressor as the final model (RMSE: 0.2094).
  - Stored in the results dictionary, accessed as results[best_model]['model'].
  - Printed confirmation: "Best Model: RandomForest with RMSE: 0.2094".
  - Likely saved the model using joblib for future predictions.
- Key Insights:
  - RandomForest's robustness and low error make it suitable for deployment.

- Model can predict sales for new games based on input features.

8. Conclusion

- Summary:
  - The pipeline effectively processed a dataset of 16,598 mobile games, handling missing values, outliers, and categorical features.
  - Four ensemble models were trained, with RandomForestRegressor achieving the best performance (RMSE: 0.2094).
  - Feature importance highlighted key predictors, aiding interpretability.
- Strengths:
  - Robust preprocessing and ensemble models handled skewed data and complex relationships.
  - Cross-validation and tuning ensured reliable performance.
- Limitations:
  - High cardinality in title_name may complicate modeling if included.
  - Inferred metrics and parameters due to incomplete notebook details.
  - residual plot) enhance understanding of data, model fit, and results.

9. Summary

- Data Exploration:

- Histograms: Numerical feature distributions (skewness, outliers).
    - Bar Plots: Frequency of categorical features.
- Preprocessing:
    - Box Plots: Pre- and post-scaling numerical feature distributions.
    - Correlation Heatmap: Feature-target relationships.
- Model Creation:
    - Parameters and requirements.
- Model Evaluation:
    - Bar Chart: Compare RMSE, MAE, R^2 across models.
    - Prediction vs. Actual Scatter Plot: Model fit visualization.
- Feature Importance:
    - Bar Plots: Top 20 features per model (already implemented, saved as 'feature_importances.png').
- Final Model:

  RandomForestRegressor.