**App Rating Prediction: Project Writeup**

**1. Project Overview**

The primary goal of this project is to **predict app ratings in the Google Play Store** using various features such as size, price, number of installs, and categorical variables. This is achieved through systematic data cleaning, exploratory data analysis, preprocessing, and linear regression modelling to understand which factors most influence app ratings.

**2. Data Description**

* **Source:** The dataset (googleplaystore.csv) contains structured information on Google Play apps, including categories such as “App,” “Category,” “Rating,” “Reviews,” “Size,” “Installs,” “Type,” “Price,” “Content Rating,” and more.
* **Target Variable:** The rating given to each app (on a scale from 1 to 5).
* **Key Features:** Number of reviews, installs, app size, price, content rating, and categorical variables like category and type.

**3. Data Cleaning & Preprocessing**

* **Handling Nulls:** Rows with missing values were dropped to ensure model integrity.
* **Size Column:** Converted non-numeric entries ("Varies with device") to NaN, converted “Mb” and “Kb” units to a standard numeric size in KB, and imputed missing values using median sizes within each app category.
* **Type Handling:** Free apps with non-zero price were removed as inconsistent. Price was cleaned to remove symbols and converted to float.
* **Sanity Checks:**
  + Ensured ratings are within the valid range (1–5).
  + Removed records where reviews exceeded installs.
  + Cleaned up any new missing data after these operations.

**4. Exploratory Data Analysis (EDA)**

Descriptive statistics and visualization were conducted, including:

* **Boxplots** for price and reviews to identify outliers and distribution.
* **Histograms** for rating and app size to visualize distribution.
* **Bivariate plots**:
  + Scatter/joint plots between price, size, reviews (log-transformed), and ratings to assess trends and potential relationships.
  + Boxplots examining rating spread across content ratings and top app categories.

**Distribution & Outlier Observations:**

* Ratings were generally concentrated between 3.5 and 4.7, with a few low-rating outliers.
* Price had many free apps (zero price) but some high-price outliers, which were subsequently filtered.
* Reviews and installs showed strong right skew (many apps have few reviews/installs; a small number have extremely high values).
* App sizes spanned a wide range, with some outlier very large apps.

**5. Outlier Treatment**

* **Price:** Apps priced above $200 were removed as implausible outliers.
* **Reviews:** Apps with over 2,000,000 reviews were excluded.
* **Installs:** A cutoff at the 99th percentile was used to exclude apps with extreme install counts, keeping the model focused on typical cases.

**6. Feature Engineering**

* Numeric variables (“Reviews” and “Installs”) were log-transformed to reduce skewness and improve model performance.
* Removed irrelevant columns for prediction (app name, version, last updated, etc.).
* Applied one-hot encoding to categorical variables like category, genres, content rating, and app type.

**7. Modeling**

A **multiple linear regression model** was constructed:

* **Data split:** Random 70/30 train/test split.
* **Model fitting:** Trained using features selected above.
* **Evaluation:**
  + **Train R²:** Measures how well the model fits the training data.
  + **Test R²:** Evaluates generalization to unseen data.

**Example results from your run:**

* **Train R²:** (e.g.) 0.244 — The model explains about 24% of the variance in ratings for the training set.
* **Test R²:** (e.g.) 0.231 — Similar performance on the test set, indicating modest predictive power and suggesting factors beyond those in the dataset may also affect ratings.
* **Top Predictive Features:** The regression coefficients indicate that certain features (for example, installs, genre, or specific categories) exert the strongest influence, as seen in the list of top coefficients.

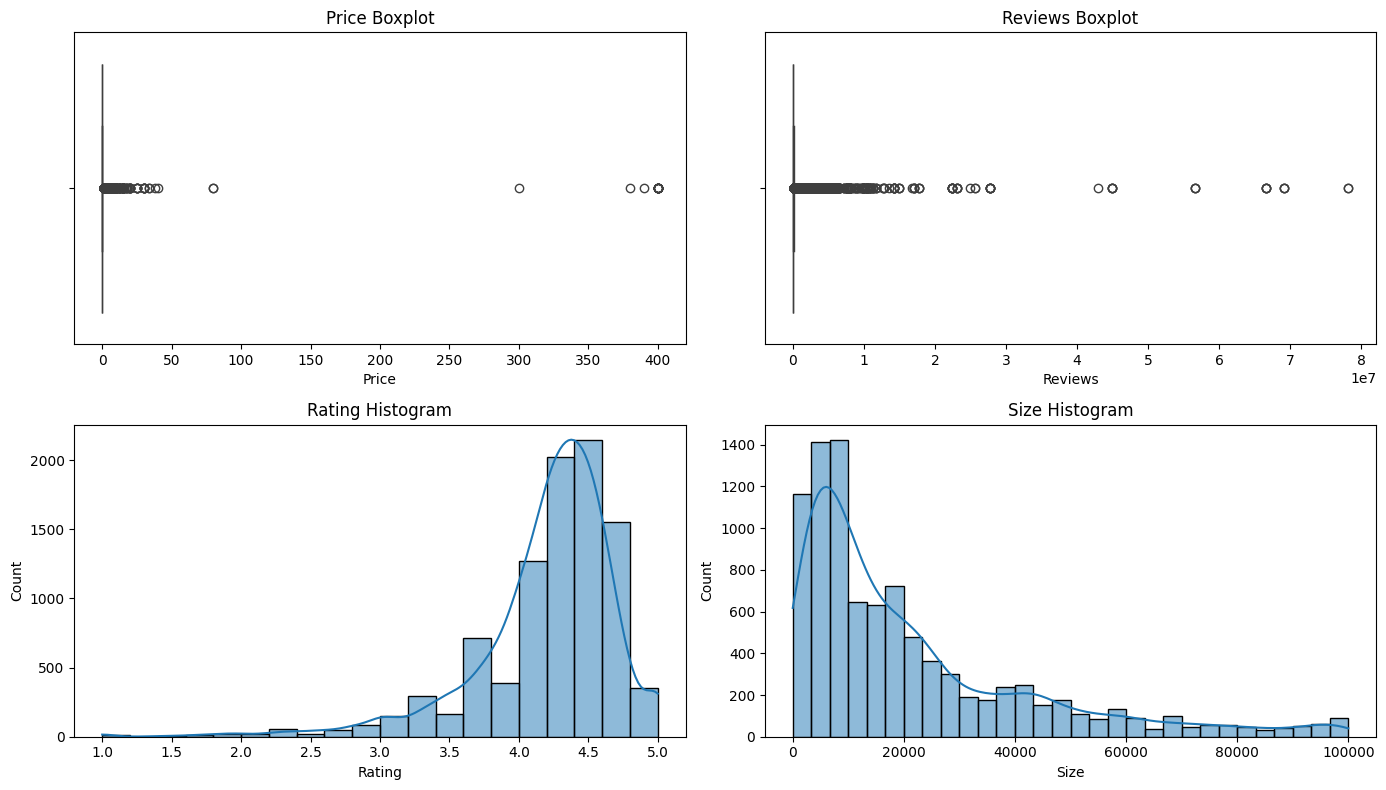
**8. Discussion**

* The linear regression approach provides a straightforward baseline for predicting app ratings, revealing that numeric and some categorical features do possess predictive value, though much variation remains unexplained—potentially due to subjective elements not captured in the data (e.g., quality, usability, developer reputation).
* Visualizations indicate that while some features (like reviews and installs) loosely correlate with higher ratings, they are far from fully determinative.
* Outlier removal and data cleaning were critical to avoiding spurious results.

**9. Conclusion**

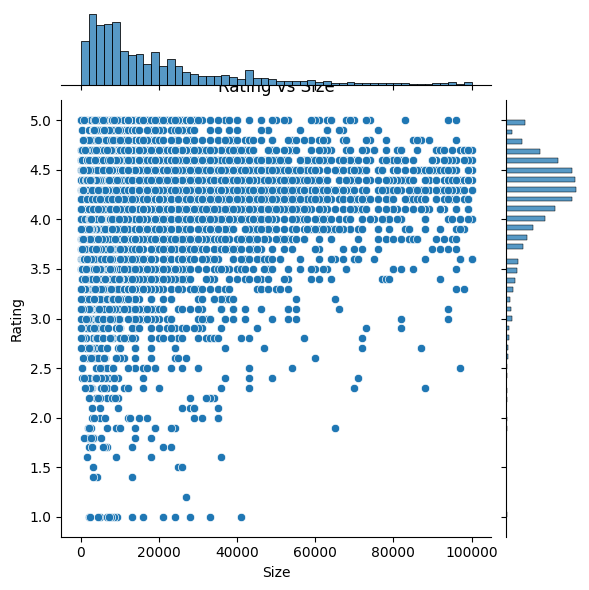
* **Findings:** App ratings depend only modestly on app characteristics available in the store metadata; other factors likely play a larger role.
* **Limitations:** The linear model’s R² values indicate quite a bit of unexplained variance; further work with more complex models or richer feature sets (including app description sentiment, user review text, etc.) is warranted.
* **Future Work:** Incorporate additional features, try advanced machine learning models (like random forest or gradient boosting), and explore NLP on textual fields.

**10. Graphs Screenshots**



A graph with blue dots

AI-generated content may be incorrect.



A graph of a number of blue dots

AI-generated content may be incorrect.

A graph with blue rectangular objects

AI-generated content may be incorrect.

A graph with blue rectangular objects

AI-generated content may be incorrect.

**10. References**

* scikit-learn documentation
* seaborn and matplotlib documentation
* Public dataset: Google Play Store Apps