Predicting User Attraction to Mobile Apps: An Analysis of the Google PlayStore Data

Prepared by 2441638

Abstract

Mobile phones are central to daily life, with 78% of the population aged 10 and over owning a device (Statista, 2024). Users rely on mobile applications for essential services such as communication, banking, healthcare and entertainment. Research indicates that 80% of mobile usage is spent on apps, with more than 250 billion apps downloaded globally(Nielsen, 2014). Applications have become essential tools, improving personal convenience while also boosting economic growth. The growing demand for mobile applications has led to a rise in application development, which plays a key role in generating significant revenue for the industry. Developers, investors, and marketers need insight into the type of applications that attract and retain users, enabling data-driven decisions for development and marketing strategies that guarantee revenue generation. The focus of this study is to understand what drives user attraction to mobile applications. Analysing the Google PlayStore platform, the number of installs apps had was used as a proxy for user interest to investigate the key factors that influence the success of the app and use them to develop a machine learning model to predict the number of installs an app could receive. The machine learning model Random Forest Regressor was used for this prediction.The study explored which app categories attract the most users, how app ratings and reviews correlate with user attraction, and whether specific features, such as social, entertainment, or productivity tools, drive higher user interest. The findings provided valuable insights to guide developers and businesses in optimising app development and marketing strategies to maximise user engagement and revenue generation.

Keywords: Machine Learning, Google PlayStore, Predictive Analytics, Application Performance, Random Forest Regressor

Contents

1. Introduction

Google Play Store and Apple's App Store are currently dominating the application (app) distribution market. Google Play Store had over 2.6 million apps available, with over 6,140 apps released daily (Statista, 2024). This vast number of apps creates a highly competitive environment for developers, despite the Play Store's 2.1 billion active android users worldwide. For developers, standing out in this crowded market is a great challenge. An effective way to measure an app's success is by analysing key metrics such as installations, ratings, and user reviews, which reflect both its performance and user perception. However, predicting an app's success pre-launch remains a difficult task.

App performance is often linked to user attraction, seen in the number of installs, reviews and ratings. As Chen (2016) explained, one-third of developer's apps get fewer than 10,000 downloads; while only 15% of apps surpass 1 million downloads (Barnard, 2014). This disparity highlights the challenge of gaining visibility in an oversaturated marketplace. While apps like Facebook boast billions of installs, most apps struggle to achieve similar success.

To predict app success pre-launch, developers can leverage machine learning (ML) techniques. By analysing important app attributes, ML models can identify patterns and trends that indicate which features will attract users. The significance of this study lies in its ability to refine predictive models for app success. A study showed that ML classifiers such as XGBoost, Random Forest, and K-Nearest Neighbors can accurately predict app performance, including ratings and installs, with up to 85.09% accuracy (Aleem and Noor, 2024a). Additionally, Aleem et al., (2024) demonstrated the effectiveness of ML predictions, achieving up to 80.34% accuracy in predicting app success.

This study used a Google Play Store dataset that is publicly available on Kaggle. Key questions included:

- What app categories attract the most users.
- Correlation of reviews and rating with user attraction.
- What features (e.g., social, entertainment) drive higher interest.

Using Random Forest Regressors, this study created and evaluated a model that can predict the volume installs an app is likely to have. This research will make a valuable contribution to the app industry by offering a predictive framework for developers to anticipate their app's success before its release helping developers to optimise their apps and increase their chances of success in a competitive market.

2. Dataset

2.1. Overview

The dataset consists of 13 features and 10,841 entries.

2.2. Data Cleaning

Data can come with garbage values that affect the model's performance, making data cleaning a crucial step. It ensures accurate and reliable results. Poor data quality can lead to misleading insights, reduced model accuracy, and biased predictions.

- **Installs:** Commas and '+' symbol were removed and "Installs" was transformed using log transform.
- **Size:** "Varies with device", was replaced with Not a Number (NaN) to represent undefined numerical values. Missing values were imputed with the median size. The app dimensions were converted to their numerical equivalent in megabytes.
- **Price:** '$' sign was removed. 'Price' was changed to integer format.
- **Categorical values:** Content Rating and Category were transformed to numerical values.
- **Rating:** There were 1474 missing entries 1. Missing entries were replaced with the median (4.3), a robust measure unaffected by outliers. [1]
- **Reviews:** 3.0M was replaced with 3000000, removing the M and transformed using log transform.
- **Current Ver** and **Android Ver:** Dropped as they were deemed irrelevant to the study.

Feature engineering involves transforming raw data into meaningful inputs to improve model performance. In this project, it included encoding categorical features ("Category" and "Content Rating") handling missing data, and creating new features such as "Free" and "Paid" from the "Type" feature. Duplicates were removed and the dataset as reduced to 9.660 unique apps.To address outliers, ratings above 5, were excluded. This normalised the data, turning skewed distributions into more symmetric ones. The transformation also reduced the scale of extreme values, improving model stability and performance.

[1] If most apps are small, but a few are unusually large, the median size reflects the central tendency better than the mean.

By addressing missing values, ensuring consistent data types, and handling outliers, the dataset was transformed into a clean, reliable form that avoids any biased predictions. These steps laid a solid foundation for building an accurate predictive model to gain meaningful insights into app data to.

3. Methodology

3.1. Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical step in the data science process, as it helps uncover patterns, trends, and relationships within the data. EDA provides an initial understanding of the dataset, ensuring that analysts make informed decisions about model selection, feature engineering, and data preprocessing. It also allows for identifying anomalies, such as missing values or outliers, which can distort results if left unaddressed.

Visualisations are a cornerstone of the EDA process. They provide an intuitive and efficient way to understand data patterns, distributions, and relationships. While raw numbers or tables can be overwhelming, visualisations make it easier to interpret and communicate insights. For mobile app analysis, visual tools can uncover trends, anomalies, and correlations that may otherwise go unnoticed.

3.2. Key Trends

The EDA revealed interesting patterns and trends that can be valuable for understanding app behaviour and making data-driven decisions. The game category is the most popular, with the highest number of installs 7. This indicates that users tend to prefer gaming apps more than other types, which is useful for developers or businesses looking to target a large audience. On the other hand, the events category has the least number of installs 7, suggesting that apps in this category may not attract as much interest from users. This insight could guide developers to either improve their event-related apps or explore other categories with higher potential. Most apps are in the ”Family” category 7, indicating its popularity. Developers aiming for success should consider targeting this category for their app placement. Analysing the density plot analysis of numerical features we can see that 2:

- **Rating:**The distribution of ratings is left-skewed. Most apps have ratings around 4 to 5. This suggests that users generally rate apps positively.
- **Reviews:**The distribution of reviews is heavily right-skewed. Most apps receive a low number of reviews, but a few apps have a significantly high number of reviews.
- **Size:**Right-skewed, indicating that most apps are relatively small in size, while a few are much larger.
- **Installs:**Extremely right-skewed, showing that the majority of apps have low installation counts, while a small number of apps dominate with very high installs.
- **Price:** Highly right-skewed, as most apps are free, with only a few high-priced apps.
- **Last Updated:**Shows a peak in recent years (2017–2018), indicating that most apps were updated during this time frame.
- **Day:**Updates across days is relatively uniform but slightly higher for specific days, showing no strong trends.
- **Month:**Most updates occur in the middle of the year, with a peak around June.
- **Year:**Shows an increase in apps released or updated in recent years, peaking sharply in 2018.

  Reviews, *Installs*, and *Size* show skewed distributions, meaning a few extreme values dominate the data. Skewed distributions can distort ML models. Normalisation ensures fair representation of all data points.

Analysing the relationship between app size and count, a bar chart revealed that the majority of applications are small in size 8. This suggests that most apps should not take up a lot of space on their devices. Lightweight apps tend to be more popular, possibly due to ease of download, faster performance, or lower resource usage.

An app’s rating is an indicator of success as it reflects user satisfaction and quality perception. A study by Sällberg and Wang and Numminen (2023), showed that apps with an average rating of 4.5 stars or more have higher install rates compared to those below 3.5 stars. The lowest rated app was ”Speech Therapy: F,” while the highest rated app was ”CS & IT Interview Questions” 4,3. The majority of apps have a rating of 4.2 5. These extremes highlight the variation in user satisfaction across different apps. Apps with low ratings might face user dissatisfaction, which could be a result of poor performance, lack of features, or bugs. On the other hand, highly rated apps generally receive positive feedback, which could indicate good functionality, user experience, or usefulness.

When looking at genres, *books and reference;Education* had the fewest installs, while *Communication* had the most 7, 7. This suggests that communication apps are in higher demand, possibly due to their social and interactive nature, which aligns with user behaviour on mobile platforms. Books and reference apps, although valuable, might not attract as many users, potentially due to the increasing popularity of more interactive or multimedia-rich categories.

There is no clear correlation between app ratings and reviews 7 [2]. This contrasts the expectation that apps with higher ratings would receive more reviews. The apps that have ratings between 4 to 4.7 have the maximum number of reviews. However, we cannot say that as the ratings increases the reviews increases. Research by Aralikatte et al., (2017), highlights this discrepancy, suggesting that star ratings and user reviews often fail to align due to biases and ambiguities in user feedback. One could propose a sentiment-based system to analyse user reviews, producing a numerical rating based on sentiment polarity. Combining this sentiment analysis with star ratings could provide a more accurate representation of app quality and reduce user confusion.

Surveys conducted by Aleem and Noor (2024b), also emphasise the need for aligning user reviews and ratings. The author notes that users often base their decision to download an app on its rating. However, inconsistencies and biases in ratings can mislead users. By incorporating sentiment analysis, a unified rating system could help users make more informed decisions.

Additional studies have explored factors influencing app success. For example, Picoto and Duarte and Pinto (2019), used multivariate analysis to show that factors (category attractiveness, diversity, and app release date) significantly impact app rankings. (Lega et al., 2022) supports this finding, noting that these factors increase the likelihood of an app being ranked among the top 50. ML techniques have also been applied showing that app ratings, content ratings, and user sentiment are critical predictors of success (Aleem and Noor, 2024a; Aleem et al., 2024). Models such as Random Forest and SVM achieved high accuracy in predicting app performance.

Overall, these key findings provide important context for building predictive models and making decisions about app development or marketing. For example, if a developer is planning to create a new app, they might consider focusing on categories where installs are higher. They could also aim for smaller app sizes to appeal to users, ensuring better performance and faster downloads. Understanding these trends helps in designing apps that align with user preferences, ultimately leading to better user engagement and higher install rates.

3.3. Feature Selection

A correlation analysis revealed that ”Reviews” had the strongest correlation with ”Installs,” making it a key feature for modelling 7, 6.

3.4. Predictive Model Selection

For this study, Random Forest Regressor (RFR) was selected as the primary predictive model for app installs, which serve as an indicator of app success. While both Random Forest (RF) and Linear Regression (LR) were considered, RFR was ultimately chosen for its superior ability to handle the complexities of the dataset and produce accurate predictions.

RF is an ensemble learning method that improves predictive accuracy by combining the outputs of multiple decision trees. Each tree in the forest is trained on a random subset of the data and features, and their predictions are averaged to generate the final output. This method reduces the risk of overfitting and increases robustness, making RF particularly effective for datasets with mixed data types, non-linear relationships, and features of varying importance. Additionally, it automatically handles feature selection and provides insights into feature importance, highlighting the most important predictors of app installs.

In contrast, LR is a simpler model that assumes a linear relationship between the input features and the target variable. It uses techniques such as Ordinary Least Squares to minimise the residual sum of squares between predicted and actual values. While computationally efficient and easy to interpret, LR relies on strict assumptions, including linearity, independence of residuals, constant variance of residuals (homoscedasticity), and normality of residuals. These assumptions limit its applicability to datasets with non-linear trends or outliers.

RF does not rely on the same strict assumptions as LR, making it more robust to outliers and noise in the dataset. Its ensemble nature reduces overfitting by averaging predictions across trees, and it can model non-linear relationships and interactions effectively. Compared to Gradient Boosting XGBoost, RF is simpler to tune, requiring fewer hyperparameter optimisations, though it may be computationally intensive when dealing with large datasets.

3.5. Preprocessing

Data preprocessing ensures that the input data is prepared for ML algorithms. Preprocessing steps,

[2] Pearson correlation (r) measures the strength and direction of a linear relationship between two variables, ranging from -1 (negative correlation) to +1 (positive correlation), with 0 indicating no linear relationship. The p-value assesses the statistical significance, showing the likelihood of observing the correlation by chance. A p-value less than 0.05 indicates a statistically significant relationship.

like feature scaling train-test splitting, and cross-validation, are crucial for building reliable and accurate model.

Feature scaling is a critical preprocessing step that ensures numerical features are adjusted to a uniform range, making them easier for ML models to interpret. In this dataset, StandardScaler was applied to standardise features, so they have a mean of 0 and a standard deviation of 1. This process is particularly important for RFR, which performs more effectively when features are on a similar scale. Without scaling, the larger values in 'Reviews' could dominate the model, leading to biased predictions. Standardising these features ensures that each variable contributes equally during training. Additionally, scaling improves the efficiency of optimisation algorithms, allowing them to converge faster and reducing training time.

By applying scaling, the dataset becomes more balanced, resulting in improved model performance and more reliable predictions.

3.6. Train Test Split

Training models is a critical step in any data science project, where the goal is to teach the model to recognise patterns in the data and make accurate predictions. This process involves splitting the dataset into two: a training and testing set. The training set "trains" the model by showing it examples of input features (e.g 'Rating', 'Size', and 'Category Encoded') and their corresponding outputs ('Log_Installs'). The model learns the relationships between the inputs and the output variable. Once the model is trained, it is tested on the unseen testing set to evaluate its ability to generalise and make predictions on new data.

Proper training techniques are essential for model accuracy and ensuring it performs well on real-world data. This includes selecting relevant features, handling missing or skewed data, and using logarithmic scaling to normalise variables. The 'Reviews' feature was transformed logarithmically to enhance model performance.

The quality of the training impacts the model's accuracy. If the training set is not representative of the broader dataset, the model may fail to capture crucial patterns, resulting in poor predictions. Likewise, improper tuning or overfitting can lead to a model that performs well during training but fails to generalise to the testing set. To address these challenges, careful pre-processing of the data, appropriate splitting, and evaluation using metrics was applied to ensure effective training and reliable predictions.

The dataset was split into 70% for training and 30% for testing. The training set fits the model, while the testing set evaluates the model's ability to generalise to unseen data. Testing on unseen data ensures the model isn't overfitting (i.e., memorising the training data rather than learning general patterns). The split is done randomly to ensure the testing set reflects the overall data distribution, and stratified sampling may be used for imbalanced datasets to maintain proportional class distributions.

While the train-test split method is straightforward and efficient, it has limitations. A single split might not represent the entire dataset's variability, especially if the dataset is small. To address this, cross-validation can be used as an alternative. Cross-validation splits the data into multiple folds and evaluates the model's performance on each fold, helping assess its robustness. However, cross-validation is more computationally intensive.

Careful consideration of data splitting, preprocessing, algorithm selection, hyperparameter tuning, and evaluation techniques ensures the model makes accurate and reliable predictions on unseen data.

4. Evaluation metrics and Model Performance

RFR was evaluated using several metrics. The evaluation scores provided insights into the model's performance and its ability to generalise to unseen data. A training R^2 score of 0.99 was achieved, indicating that the model captured 99% of the variance in the training data. The model effectively learned patterns and relationships within the dataset. However, a high R^2 score raises the concern of potential overfitting. Further validation such as hyperparameter tuning or cross-validation would be required to ensure that the model is generalising well to unseen data. The validation R^2 score of 0.94, in conjunction with the out-of-bag (OOB) score of 0.94, confirms that the model generalises effectively. The OOB score is a robust internal validation metric specific to RF. It showcases the model's ability to predict unseen data during training using samples excluded from bootstrap datasets. The close alignment of both the validation R^2 and the OOB scores reinforces the model's reliability in predicting outcomes on unseen data.

The Mean Absolute Error (MAE) of 0.775 and the Root Mean Absolute Error (RMAE) of 0.881 highlight the model's reliability and accuracy. The MAE is indicative of the model's average predictions deviating from the actual values by less than one unit; The RMAE, which penalises larger errors more, demonstrates a consistent and reliable prediction across the dataset. Research on software performance prediction using RFR concluded that RF models can capture complex relationships in software metrics, achieving high predictive accuracy (P, 2016). This aligns with this current study, where RFR captured 99% of the variance in training data and 94% in test data, demonstrating its capability to model intricate patterns.

Additionally, a comparative study that examined the performance of RF and logistic regression models on banking marketing data found that RF out performed logistic regression, achieving superior accuracy and generalisation (Varol Malkoçoğlu and Utku Malkoçoglu, 2020). Another study investigating house price predictions using RFR reported a R^2 score of 0.85, with a RMSE of 0.1523 and MAE of 0.1132 (Mao, 2024). Although the R^2 score is slightly lower than the score achieved in this study, the strong performance metrics in both cases emphasise the predictive strength of RF models. The differences could be attributed to variations in dataset size and complexity. Overall, the performance of RFR model is consistent to those reported in literature. The alignment of evaluation metrics and low error rates with benchmark studies underscore the robustness and reliability of RF models for predicting app installs and confirm their broader applicability in handling complex datasets across diverse domains.

5. Limitations

Categories with fewer apps may produce skewed results, potentially affecting the model's ability to generalise across all app categories. The dataset lacked external features such as marketing budgets or seasonal trends, which may influence app installs and could provide a more comprehensive understanding of the factors driving app success.

6. Recommendations

Incorporating additional data sources, such as user demographics or advertising data, would provide a better understanding of factors influencing app installs, enhancing future analysis. To improve model accuracy, advanced hyperparameter tuning could be used. The growth in the "Health & Fitness" & "Education" category, highlights increasing user interest in self-improvement apps. Developers could focus on this area of apps for future success. Leveraging AI-driven personalisation features could potentially drive app popularity in the coming years, making it a valuable strategy for app developers and marketers.

7. Conclusion

This study analysed app installation trends on the Google Play Store to predict user attraction and identify key factors driving app success. RFR was selected as the primary predictive model due to its ability to handle non-linear relationships, mixed data types, and feature interactions effectively.

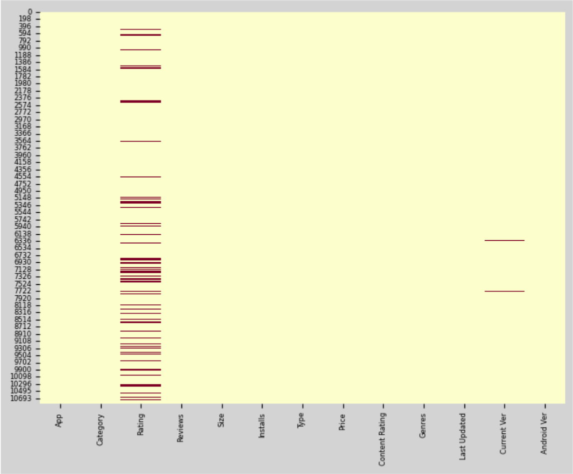


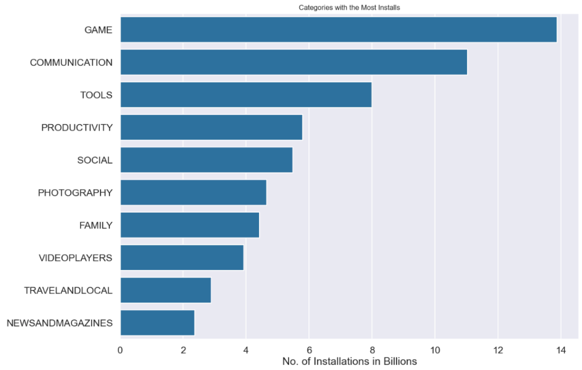
Figure 1: Features with Missing Values

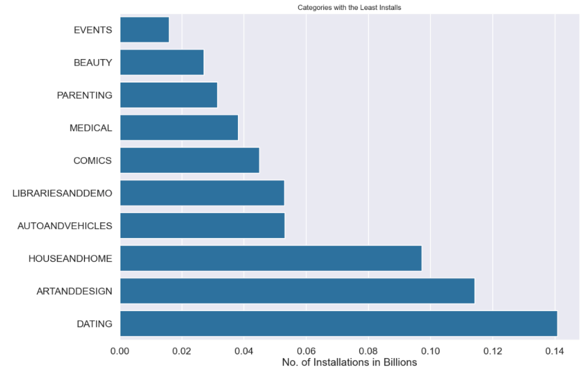
Evaluation metrics demonstrated that the model performed exceptionally well, with high R² scores for both training and validation data, indicating robust predictive accuracy and generalisation capability.

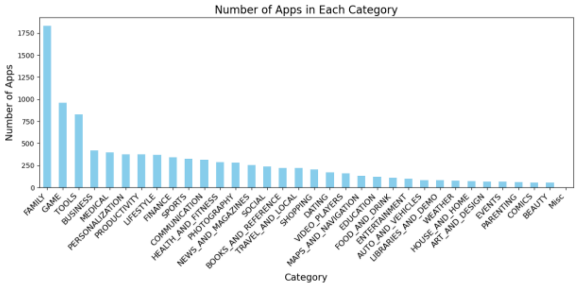
The analysis revealed that reviews primarily correlated and influenced install numbers. Games and Social categories had high user attraction, while "Health & Fitness" & "Education" indicated growing interest in self-improvement apps. The importance of free pricing and regular app updates further emphasised user preferences in app adoption.

Despite the model's strong performance, limitations such as data imbalance and the absence of external factors (marketing budgets or seasonal trends) were identified. These challenges can be addressed in future studies by incorporating additional data and advanced techniques to improve predictive accuracy and provide deeper insights.

This study demonstrates the value of data-driven decision-making for app developers and marketers. By understanding trends and optimising key factors, businesses can enhance app visibility, cater to user preferences, and ultimately achieve greater success in the competitive app marketplace.







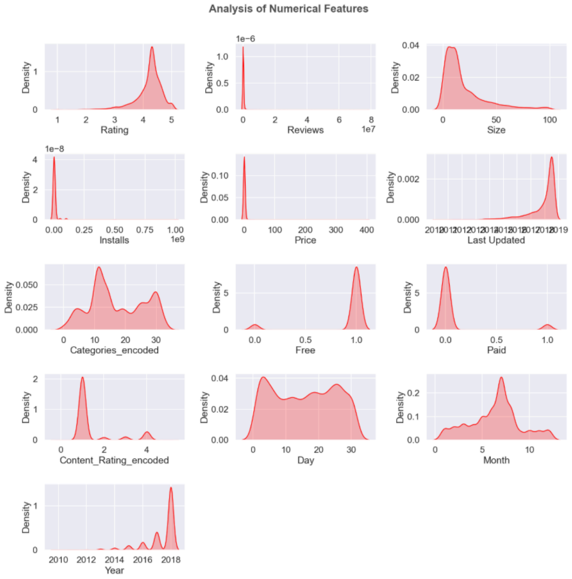
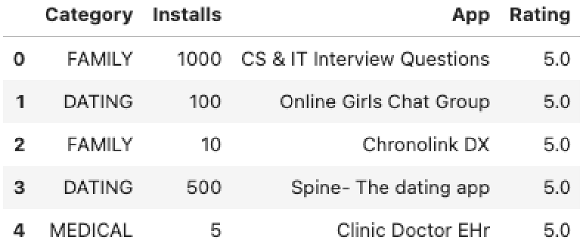
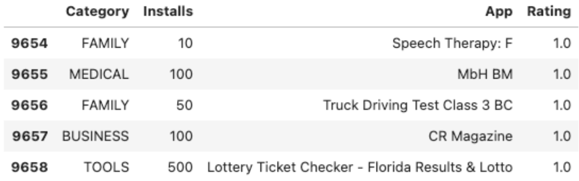


Figure 2: Density Plot Analysis of Numerical Features in the Dataset



	Category	Installs	App	Rating
0	FAMILY	1000	CS & IT Interview Questions	5.0
1	DATING	100	Online Girls Chat Group	5.0
2	FAMILY	10	Chronolink DX	5.0
3	DATING	500	Spine- The dating app	5.0
4	MEDICAL	5	Clinic Doctor EHr	5.0

Figure 3: Top 5 Apps with 5 Star Rating



	Category	Installs	App	Rating
9654	FAMILY	10	Speech Therapy: F	1.0
9655	MEDICAL	100	MbH BM	1.0
9656	FAMILY	50	Truck Driving Test Class 3 BC	1.0
9657	BUSINESS	100	CR Magazine	1.0
9658	TOOLS	500	Lottery Ticket Checker - Florida Results & Lotto	1.0

Figure 4: Top 5 Apps with 1 Star Rating

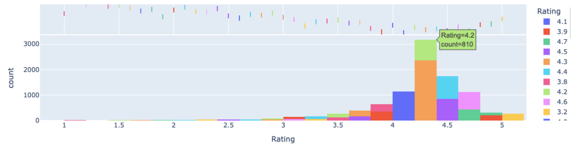
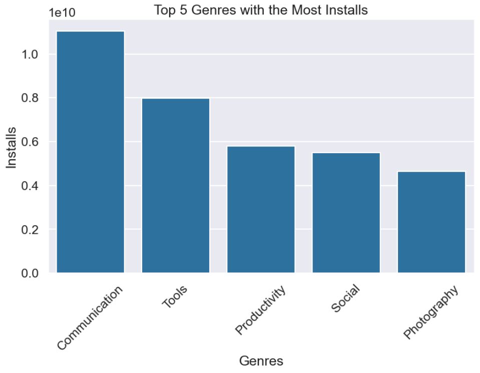
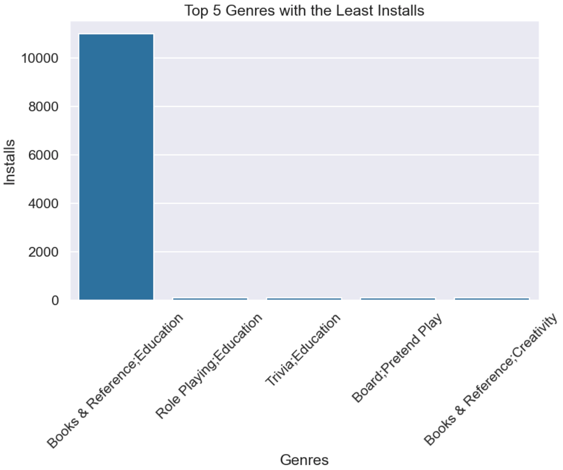
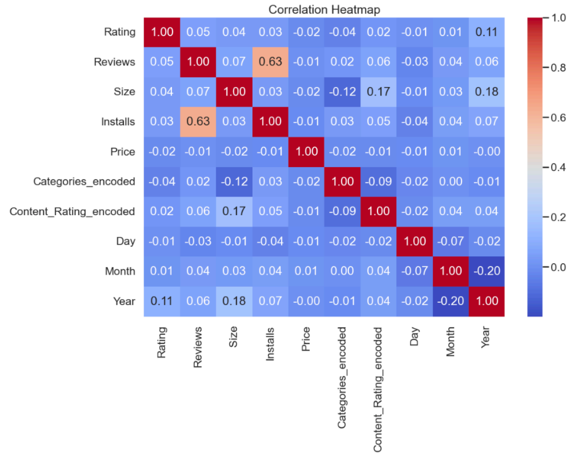


Figure 5: The Number of Apps with Ratings







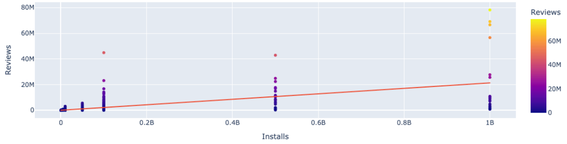
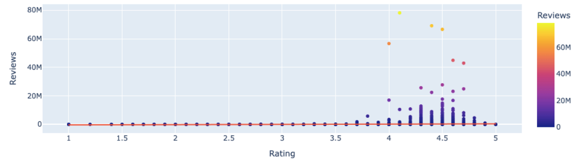


Figure 6: Correlation between Installs and Reviews



```
Pearson Correlation: 0.050
P-value: 0.00000079
```

Figure 7: Correlation between Rating and Reviews

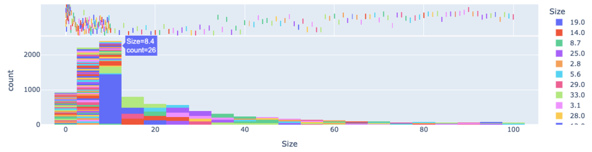
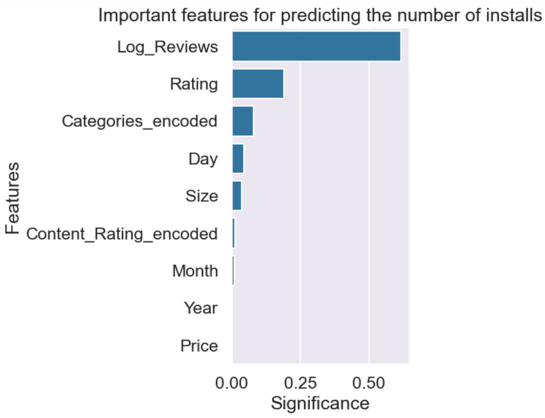


Figure 8: App Sizes



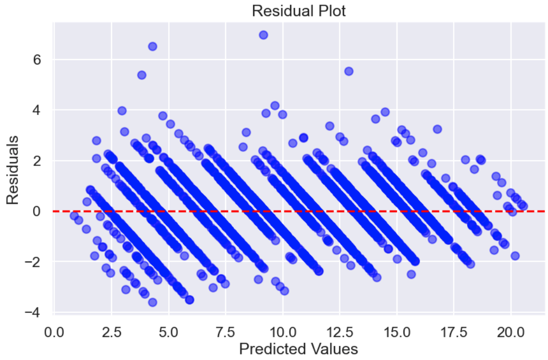


Figure 9: Points are evenly distributed around zero, indicating a well-calibrated model. The funnel shape suggests heteroscedasticity, where prediction errors increase for larger values, hinting at difficulty with higher-value predictions.

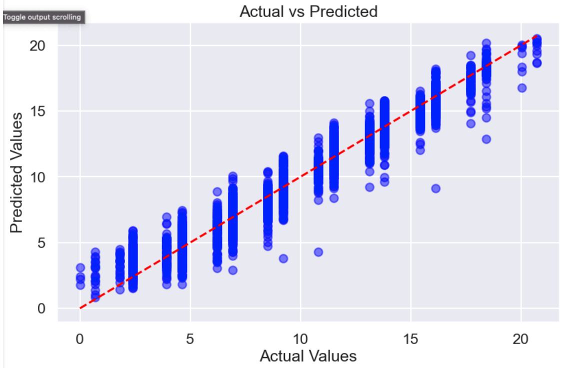


Figure 10: Most points cluster near the diagonal, indicating reasonable accuracy. Deviations reveal prediction errors.

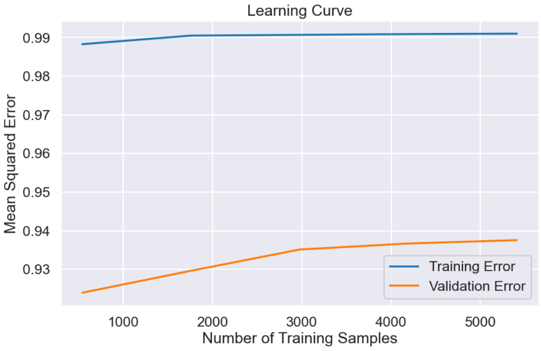


Figure 11: Low training error indicates a good fit and low bias. Rising validation error suggests diminishing returns with more data. The gap between the curves points to slight overfitting but acceptable generalisation to unseen data.

Bibliography

Muhammad Aleem, Ehtesham Ali, Mohd Abdullah Al Mamun, Jalal Uddin Md Akbar, Khadeeja Saeed, and Aqsa Saleem. 2024. Harnessing ensemble learning approaches for strong mobile app success prediction model. *International Journal of Communication Networks and Information Security*, 16(4):1414–14225.

Muhammad Aleem and Noorhuzaimi Mohd Noor. 2024a. Predicting android app success before google play store launch using machine learning. *Library Progress International*, 44(3):1907–1918.

Muhammad Aleem and Noorhuzaimi Mohd Noor. 2024b. Predicting android app success before google play store launch using machine learning. *Library Progress International*, 44(3).

Rahul Aralikatte, Giriprasad Sridhara, Neelamadhav Gantayat, and Senthil Mani. 2017. Fault in your stars: An analysis of android app reviews. *arXiv preprint arXiv:1708.04968*.

David Barnard. 2014. Shaping the app store. https://davidbarnard.com/post/73439313124/shaping-the-app-store. Accessed: 30 December 2024.

Andrew Chen. 2016. New data shows why losing 80% of your mobile users is normal, and that the best apps do much better. https://andrewchen.com/new-data-shows-why-losing-80%-of-your-mobile-users-is-normal-and-why-the-best-apps-do-better/. Accessed: 30 December 2024.

Mathieu Lega, Corentin Burnay, and Stephane Faulkner. 2022. Predicting the rating of an app beyond its functionalities: Introducing the app publication strategy. *Lecture Notes in Business Information Processing*.

Mohan Mao. 2024. A comparative study of random forest regression for predicting house prices using. *ResearchGate*, 85:969–974.

Nielsen. 2014. Smartphones: So many apps, so much time. https://www.nielsen.com/insights/2014/smartphones-so-many-apps-so-much-time/. Accessed: 30 December 2024.

Sudhakar P. 2016. Software performance prediction using random forest-based regression analysis. *ResearchGate*.

Winnie Ng Picoto, Ricardo Duarte, and Inês Pinto. 2019. Uncovering top-ranking factors for mobile apps through a multimethod approach. *Journal of Business Research*, 101:668–674.

Statista. 2024. Forecast number of mobile users worldwide 2020-2025. https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/. Accessed: 30 December 2024.

H Sällberg, S Wang, and E Numminen. 2023. he combinatory role of online ratings and reviews in mobile app downloads: an empirical investigation of gaming and productivity apps from their initial app store launch. *Journal of Marketing Analytics*, 11:426–442.

Ayşe Berika Varol Malkoçoğlu and Şevki Utku Malkoçoglu. 2020. Comparative performance analysis of random forest and logistic regression algorithms. In *2020 5th International Conference on Computer Science and Engineering (UBMK)*, pages 25–30.