Predicting App Success on Google Play Store

BY: LAUREL STEWART, YUNJIU LI

Objective and Motivation

Goal: Predict App Success (installs per month) using app level data from google play store

Motivation:

1. Android dev cost = high, revenue = uncertain.
2. Users spend lesser than iOS users.
3. Device fragmentation makes support costly.
4. Use data from Google Play Store to understand what drives app success.

Data Collection

- **Data Source:** Combination of preexisting datasets (Kaggle) and custom web scraping using Apify and google-play-scraper, originally more than 40 columns.
- **Scope:**

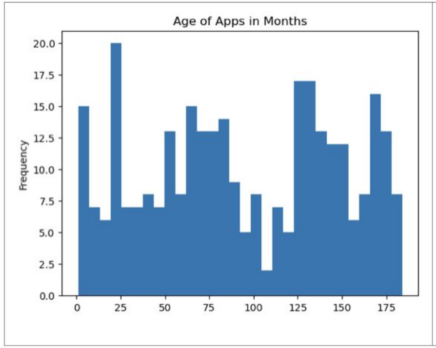
1. 316 apps across 10 popular categories.
2. Excluded: Google-owned apps, major media brands, social networks

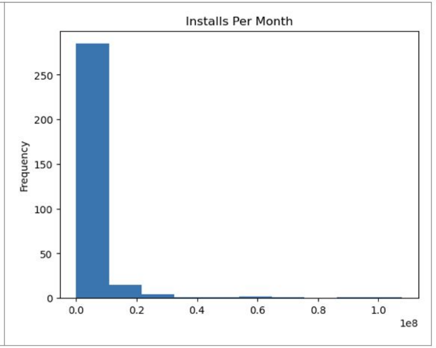
- **Dataframe:** 27 columns, cleaned and encoded.

Removed in processing	Feature	Description
	title	App title/name
	description	Full app description
	descriptionHTML	HTML formatted description
	summary	Brief app summary
x	installs	Install count range
x	minInstalls	Minimum number of installs
	realInstalls	Actual install count
	score	App rating score
	ratings	Number of ratings
	reviews	Number of reviews
X	histogram	Rating distribution histogram
	price	App price
	free	Whether app is free
	currency	Price currency
	sale	Whether app is on sale
	saleTime	Sale duration/timing
	originalPrice	Original price before sale
	saleText	Sale description text
	offersIAP	Offers in-app purchases
	inAppProductPrice	In-app purchase pricing
x	developer	Developer name
x	developerId	Developer identifier
x	developerEmail	Developer contact email
x	developerWebsite	Developer website URL
x	developerAddress	Developer address
x	privacyPolicy	Privacy policy URL
	genre	App genre/category
	genreId	Genre identifier
	categories	App categories
x	icon	App icon image
x	headerImage	Header/banner image
x	screenshots	App screenshot images
x	video	Promotional video
x	videoImage	Video thumbnail image
	contentRating	Age/content rating
	contentRatingDescription	Content rating details
	adSupported	Whether app shows ads
	containsAds	Contains advertisements
	released	Release date
	lastUpdatedOn	Last update date
X	updated	Update timestamp
	version	App version number
X	comments	User comments/reviews
x	appId	Unique app identifier
x	url	Google Play Store URL

Data Cleaning and Pre-Processing

- **Removed columns(Missing Values):**
  - redundant or unavailable data (e.g., sale time and original price).
- **Feature adjustments :**
  - Create InstallsPerMonth to normalize across different app ages
- **Boolean & Categorical Conversion:**
  - Convert T/F -> 1/0
  - One-hot encoded genre and other categories





Feature Engineering

- **Key Features Created**

1. Installs per month = installs/age.
2. App age(in months),
3. Update frequency

- **Format Changes**

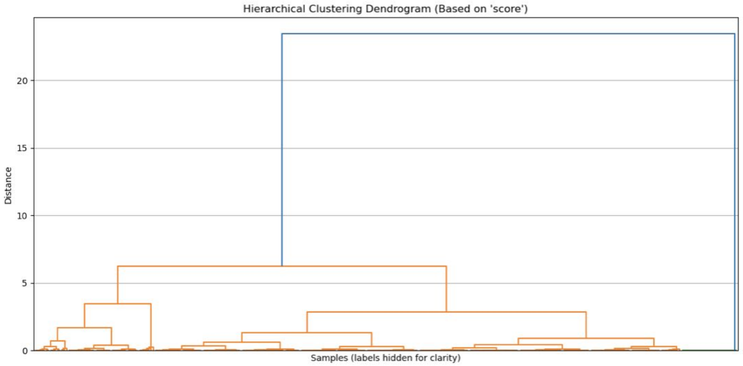
1. Boolean -> 0/1,
2. genre -> dummy variables,

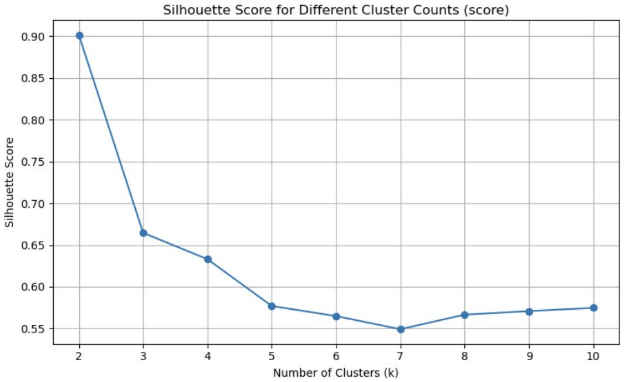
- **Tried but removed**

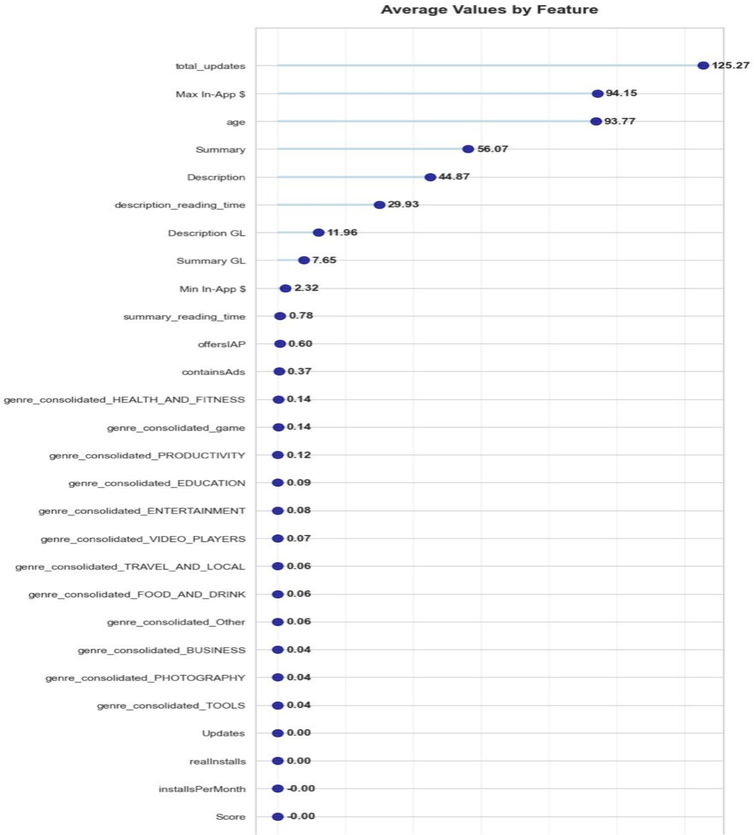
1. Readability scores of descriptions
2. Clustering-based features (didn’t help)

```python
# Feature Engineering
df['log_installsPerMonth'] = np.log1p(df['installsPerMonth'])  # Log-transformed target
df['iap_price_range'] = df['maxInAppProductPrice'] - df['minInAppProductPrice']  # Price range
df['score_to_age_ratio'] = df['score'] / (df['age'] + 1)  # Score normalized by age
df['readability_blend'] = (df['description_readability_score'] + df['summary_readability_score']) /
2  # Combined readability
df['update_density'] = df['total_updates'] / (df['age'] + 1)  # Updates per unit age
df['score_reading_interact'] = df['score'] * df['description_reading_time']  # Interaction term
```

Feature Engineering



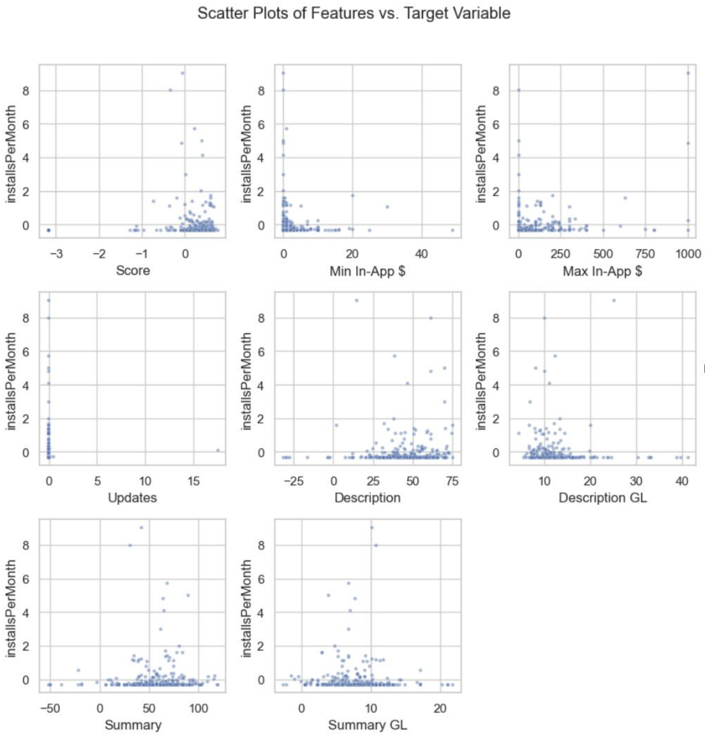




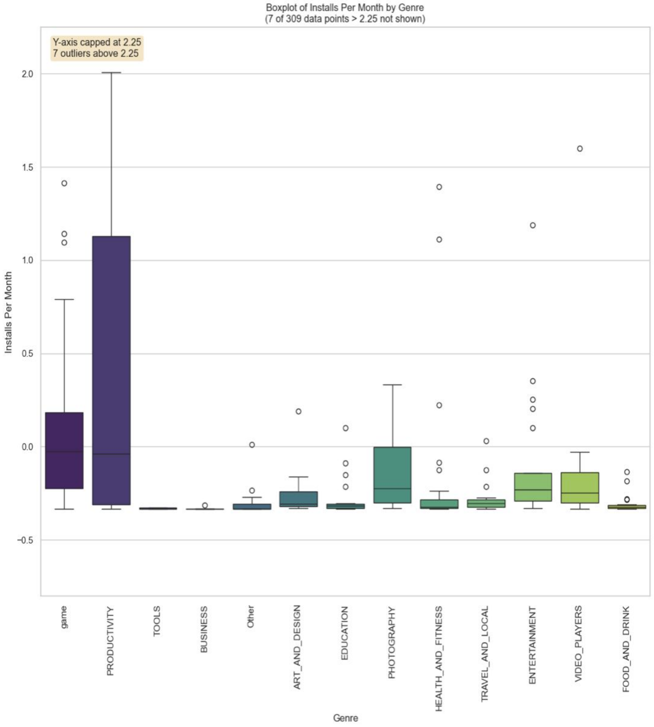
Data Analysis

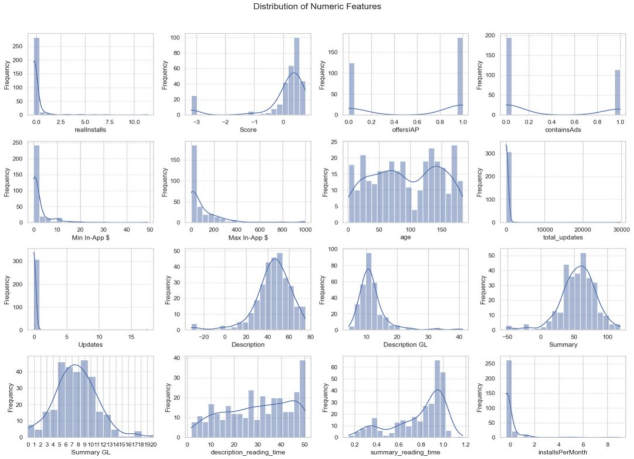
- **Visuals:**
  - Histograms of installs/month.
  - Box plots of installs by genre.

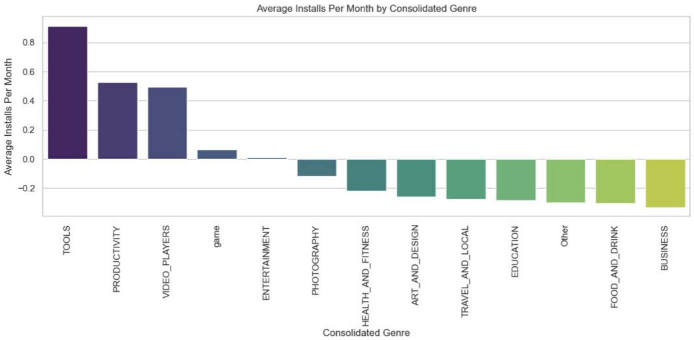
- **Key Insights:**
  - Target variable highly skewed.
  - Significant variance across genres.
  - Correlations revealed data leakage (realInstalls).



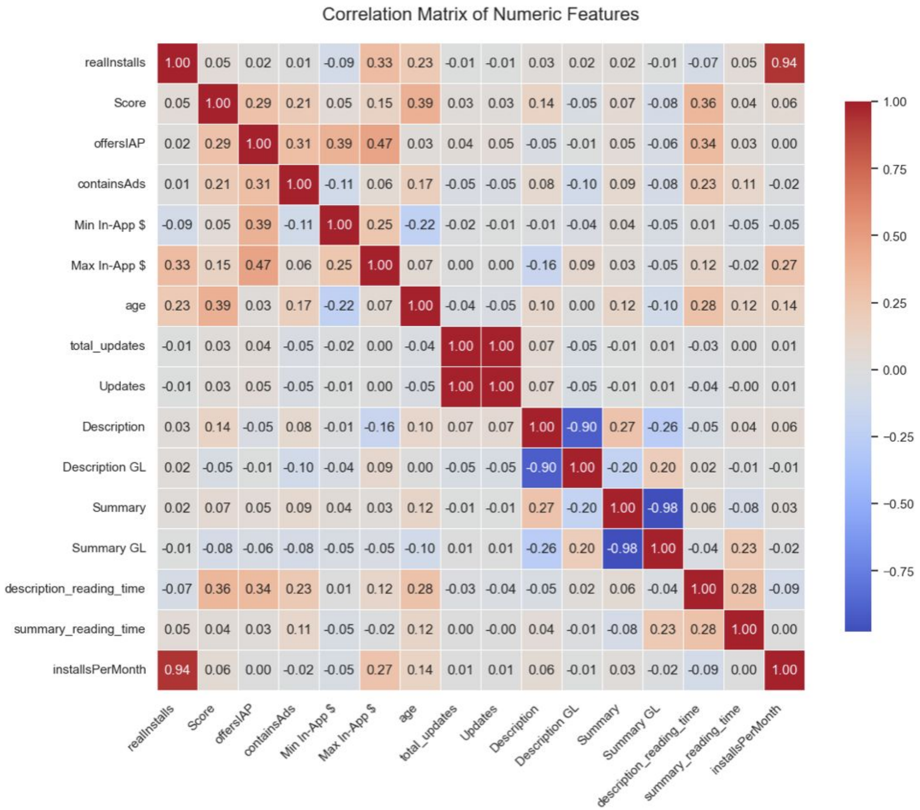
Data Analysis







Data Analysis



Modeling Approaches

Models Tested:

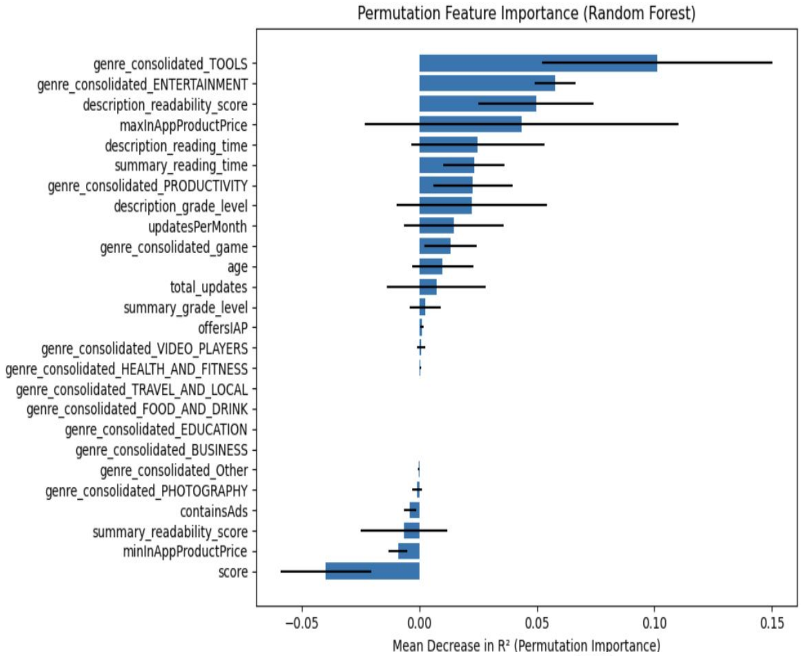
- Linear Regression
- Random Forest Regressor
- Gradient Boosting Regressor

Initial Results (with leakage):

- Unrealistically high accuracy due to realInstalls included.

After Fixing:

- Gradient Boosting performed best (R^2 = 0.1112).
- Linear & RF underperformed ($R^2 < 0$).



Model	**R^2 Score**	**RMSE**
Linear Regression	-0.0812	1.1733
Random Forest Regressor	-0.1983	1.2352
Gradient Boosting Regressor	**0.1112**	**1.0638**

Key Findings

- Most apps don’t get many downloads.
- App genre affects installs, but not in a clear or consistent way.
- Apps with frequent updates and in-app purchases tend to do better.
- Readability of the app description doesn’t seem to matter.
- Some variables (like total installs) can accidentally give away the answer and hurt the model.
-

Conclusions & Lesson Learned

- Web-scraped data can be messy—cleaning and validation are essential.
- Some engineered features added noise instead of value.
- To predict real success, we need more context—like marketing budgets or team size.
- Big app hits are hard to predict with limited features alone.
-

Thanks!