Google Play Store Apps

Predicting Apps Rating for the Android Market

STAT GR5243 Final Project

Stella Dai, Liang Zhao, Siwei Chen, Wenwei Kuang

Project 1 Review

Analyzing adverse drug reactions on patients

- In project 1, we have used FDA's Adverse Drug Events Database to explore the side effects and ADRs among the global FDA-approved drugs.
- Our business goal is to investigate the adverse reactions experienced by patients and thus boost medication safety.
- Our final dataframe contains 12578 observations and 14 variables. We aim to use patient age, sex, and type of drugs intake to predict the seriousness level of adverse reaction.

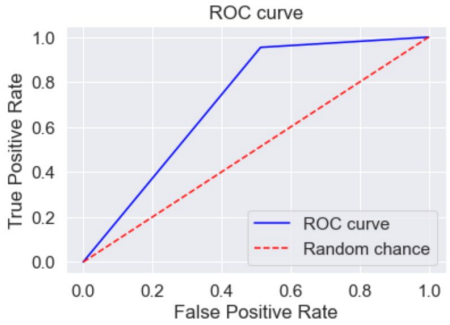
	serious_results	age_label	patientsex	drug_HUMIRA	drug_DUPIXENT	drug_INBRIJA	drug_NURTEC ODT	drug_SKYRIZI	drug_RINVOQ	drug_PREDNISONE
3	1	4	0	0	0	0	0	0	0	0
8	0	4	1	0	0	0	0	0	0	0
9	1	4	0	0	0	0	0	0	0	0
10	1	5	0	0	0	0	0	0	0	0
12	0	2	1	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
25994	1	4	1	0	0	0	0	0	0	0
25995	1	4	1	0	0	0	0	0	0	0
25996	1	5	0	0	0	0	0	0	0	0
25997	1	4	1	0	0	0	0	0	0	0
25999	0	5	1	0	0	0	0	0	0	0

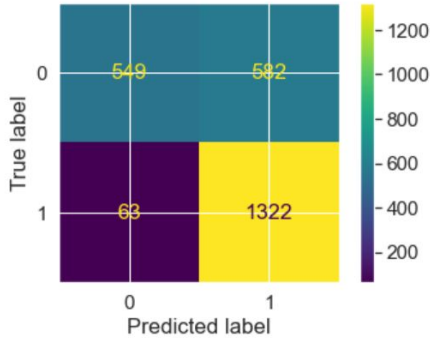
12578 rows × 14 columns

Final Model selection

Logistic Regression

- High accuracy: 0.7345
- Low false positive cases: 63/2516 = 0.025
- All variables are significant in terms of p-values.
- Patients might want to pay extra attention to those drugs that cause significant side effects. (i.e. PREDNISONE, ASPIRIN)





Logit Regression Results

Dep. Variable:	serious_results	**No. Observations:**	10062
Model:	Logit	**Df Residuals:**	10049
Method:	MLE	**Df Model:**	12
Date:	Wed, 22 Mar 2023	**Pseudo R-squ.:**	0.2053
Time:	18:14:55	**Log-Likelihood:**	-5501.6
converged:	True	**LL-Null:**	-6923.3
Covariance Type:	nonrobust	**LLR p-value:**	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
age_label	0.0918	0.022	4.177	0.000	0.049	0.135
patientsex	-0.2950	0.047	-6.242	0.000	-0.388	-0.202
drug_HUMIRA	-1.9641	0.119	-16.516	0.000	-2.197	-1.731
drug_DUPIXENT	-2.4945	0.167	-14.946	0.000	-2.822	-2.167
drug_INBRIJA	-5.1725	0.585	-8.837	0.000	-6.320	-4.025
drug_NURTEC ODT	-4.5306	0.508	-8.919	0.000	-5.526	-3.535
drug_SKYRIZI	-2.2128	0.228	-9.691	0.000	-2.660	-1.765
drug_RINVOQ	-1.3100	0.191	-6.848	0.000	-1.685	-0.935
drug_PREDNISONE	1.6162	0.171	9.454	0.000	1.281	1.951
drug_RUBRACA	-1.5481	0.245	-6.314	0.000	-2.029	-1.068
drug_PAXLOVID	-2.2874	0.158	-14.473	0.000	-2.597	-1.978
drug_ASPIRIN	0.9466	0.167	5.666	0.000	0.619	1.274
other_drug	0.4967	0.094	5.287	0.000	0.313	0.681

Introduction for Project 2

Motivation

- In this project, we will be using the dataset of Google Play Store apps to help app-making businesses.
- By understanding the factors that influence app ratings can help app developers create better products. By analyzing user reviews, developers can identify areas for improvement and make changes to increase user satisfaction.
- In order to achieve those goals, we will conduct effective machine learning models to analyze and predict the app ratings based on category and price the app, number of user reviews, install amount, and target age group.
- Also, we will conduct sentiment analysis based on the users' reviews of apps to predict the type of sentiment.
- Overall, our project aim to provide valuable insights for both app developers and consumers, leading to better app products and more satisfying user experience.

EDA

Overview

- Overview of the dataset
- Choose the attributes we want to further analysis

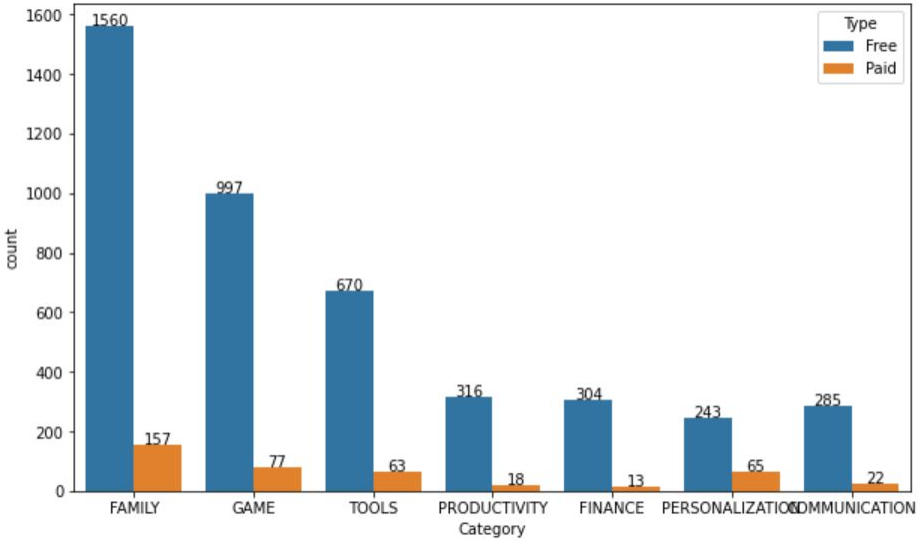
	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10000.0	Free	0.0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500000.0	Free	0.0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.7	87510	8.7M	5000000.0	Free	0.0	Everyone	Art & Design	August 1, 2018	1.2.4	4.0.3 and up
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.5	215644	25M	50000000.0	Free	0.0	Teen	Art & Design	June 8, 2018	Varies with device	4.2 and up
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.3	967	2.8M	100000.0	Free	0.0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10353	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5000.0	Free	0.0	Everyone	Education	July 25, 2017	1.48	4.1 and up
10354	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100.0	Free	0.0	Everyone	Education	July 6, 2018	1.0	4.1 and up
10355	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1000.0	Free	0.0	Everyone	Medical	January 20, 2017	1.0	2.2 and up
10356	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1000.0	Free	0.0	Mature 17+	Books & Reference	January 19, 2015	Varies with device	Varies with device
10357	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10000000.0	Free	0.0	Everyone	Lifestyle	July 25, 2018	Varies with device	Varies with device

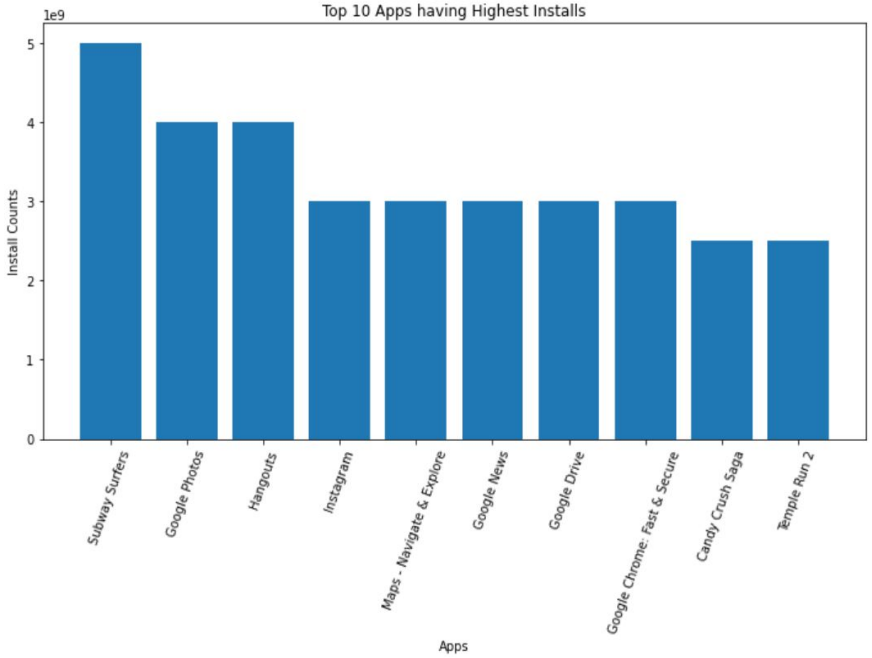
10357 rows × 13 columns

EDA

Overview and Data Wrangling

- Distribution with Top 7 Category (Free vs Paid)
- Top 10 Apps having Highest Installs

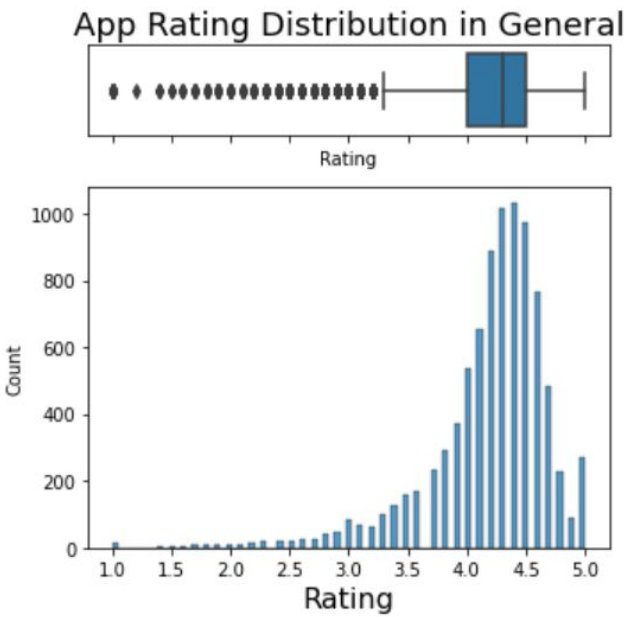


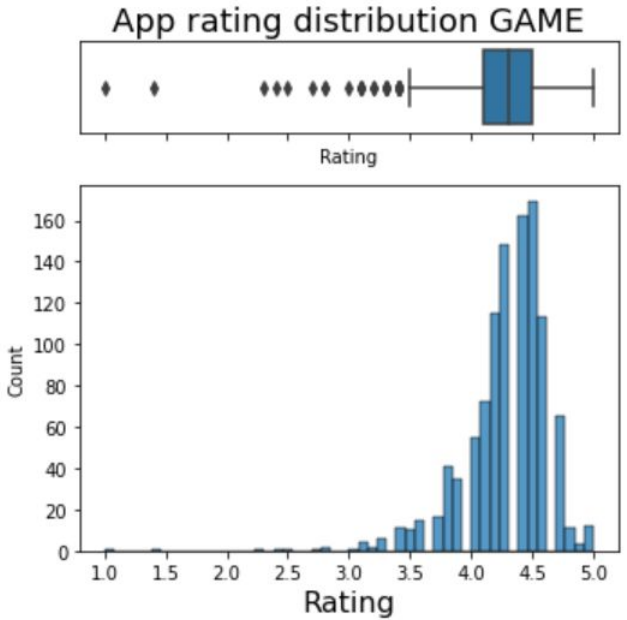


EDA

Overview and Data Wrangling

- Rating Distribution Based on Rating in General & Popular Reviews

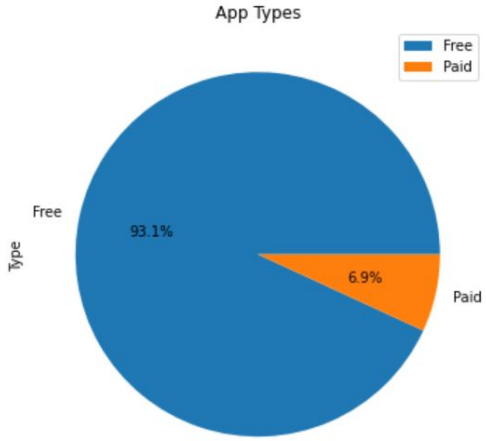




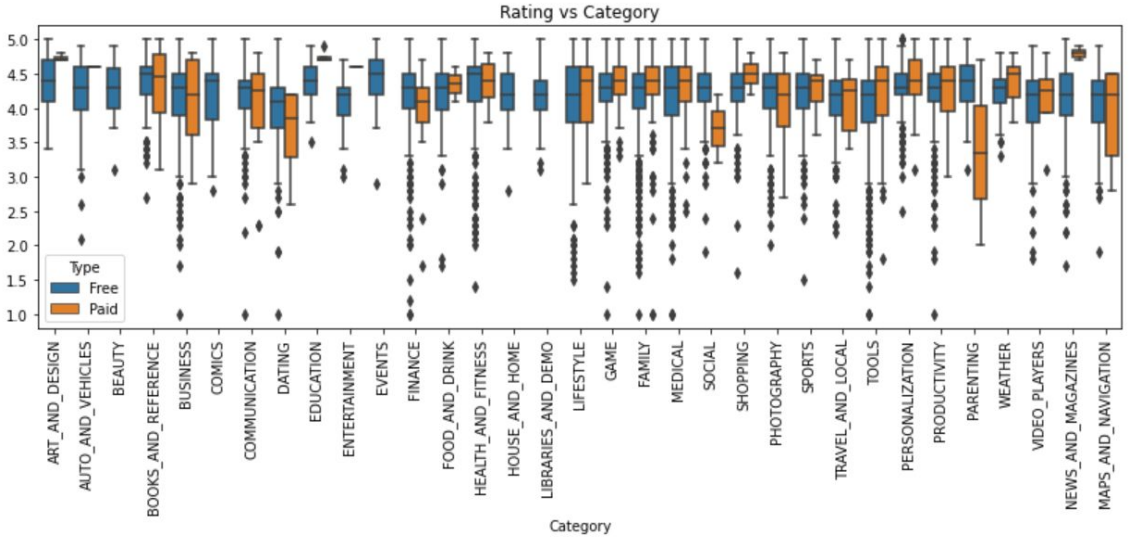
EDA

Overview and Data Wrangling

- Percentage of Paid vs. Free type in Google Play Store



- Which Category has higher rating in general?
- From the graph, the blue bars (Free) has higher rating, comparing to free which has higher difference in the rating range.



EDA

Check assumptions

- The correlation between predictor variables are very low, except ‘reviews’ and ‘installs’, so there may be collinearity issues
- Low correlation between predictor variables and response variable “Rating”

Data Preprocessing

```
df.drop_duplicates(inplace=True, ignore_index=True)
df = df[df.Category != '1.9']
df=df.dropna()
df["Price"]=df['Price'].str.replace('$','').astype(float)
df['Installs'] = df['Installs'].str.replace('+','').str.replace(',','').astype(float)
df[~df.Reviews.str.isnumeric()]
df['Reviews'] = pd.to_numeric(df['Reviews'])
df2=pd.get_dummies(df.Category,prefix='Category')
df= df.drop(columns=['Size','App', 'Category','Type','Content Rating','Genres','Last Upd
data = df.join(df2)
```

Data cleaning:

- Check for duplicates and remove duplicate rows
- Remove missing values for predictor variables and the unnecessary columns
- Create dummy variables for category of APPs using one-hot encoding
- Standardize training data using MinMaxScaler to prevent sensitive issue

Train_test_split:

```
data['rate_classify'] = pd.Series(0)
for ind in data.index:
    if data['Rating'][ind]>=4.5:
        data['rate_classify'][ind]=1
    elif data['Rating'][ind]<4.5:
        data['rate_classify'][ind]=0
```

- For regression, keep rating as continuous value.
- For classification, assign rating over 4.5 as positive rate; below 4.5 as negative rate.
- Then stratified sampling based on the distribution of rating.

Data Preprocessing

Final Dataframe for Modeling

- Finally, our dataframe contains 8886 observations and 4 features(not including dummy variables).
- We aim to use category, price of the app, number of user reviews, install counts, and target age group to predict the rating of APPs.

	Category	Rating	Reviews	Installs	Price	Category_ART_AND_DESIGN	Category_AUTO_AND_VEHICLES	Category_BEAUTY	Cat
0	ART_AND_DESIGN	4.1	159	10000.0	0.0	1	0	0	
1	ART_AND_DESIGN	3.9	967	500000.0	0.0	1	0	0	
2	ART_AND_DESIGN	4.7	87510	5000000.0	0.0	1	0	0	
3	ART_AND_DESIGN	4.5	215644	50000000.0	0.0	1	0	0	
4	ART_AND_DESIGN	4.3	967	100000.0	0.0	1	0	0	
...	...	...	...	...	...	...	...	...	
10350	FAMILY	4.0	7	500.0	0.0	0	0	0	
10352	FAMILY	4.5	38	5000.0	0.0	0	0	0	
10353	FAMILY	5.0	4	100.0	0.0	0	0	0	
10355	BOOKS_AND_REFERENCE	4.5	114	1000.0	0.0	0	0	0	
10356	LIFESTYLE	4.5	398307	10000000.0	0.0	0	0	0	

8886 rows × 38 columns

Data Modeling

Linear Regression

	coef	std err	t	P>|t|	[0.025	0.975]
const	4.0686	0.009	463.957	0.000	4.051	4.086
x1	0.6454	0.203	3.179	0.001	0.247	1.043
x2	0.1340	0.089	1.498	0.134	−0.041	0.309
x3	−0.2822	0.146	−1.929	0.054	−0.569	0.005
x4	0.3310	0.072	4.606	0.000	0.190	0.472
x5	0.0917	0.069	1.326	0.185	−0.044	0.227
x6	0.1871	0.086	2.175	0.030	0.018	0.356
x7	0.2626	0.043	6.137	0.000	0.179	0.347
x8	0.0388	0.037	1.063	0.288	−0.033	0.110
x9	0.1536	0.076	2.028	0.043	0.005	0.302
x10	0.0391	0.034	1.159	0.246	−0.027	0.105
x11	−0.0957	0.045	−2.119	0.034	−0.184	−0.007
x12	0.3159	0.049	6.382	0.000	0.219	0.413
x13	0.0366	0.055	0.660	0.510	−0.072	0.145
x14	0.3288	0.080	4.088	0.000	0.171	0.486
x15	0.1224	0.016	7.619	0.000	0.091	0.154
x16	0.0728	0.032	2.302	0.021	0.011	0.135
x17	0.0724	0.055	1.328	0.184	−0.034	0.179
x18	0.1962	0.019	10.175	0.000	0.158	0.234
x19	0.1693	0.036	4.688	0.000	0.098	0.240
x20	0.0619	0.068	0.911	0.362	−0.071	0.195
x21	0.0847	0.067	1.259	0.208	−0.047	0.217
x22	0.0506	0.033	1.543	0.123	−0.014	0.115
x23	−0.0002	0.049	−0.005	0.996	−0.097	0.097
x24	0.1249	0.033	3.732	0.000	0.059	0.190
x25	0.0653	0.040	1.652	0.099	−0.012	0.143
x26	0.1813	0.081	2.226	0.026	0.022	0.341
x27	0.2590	0.033	7.804	0.000	0.194	0.324
x28	0.1217	0.033	3.732	0.000	0.058	0.186
x29	0.1017	0.033	3.122	0.002	0.038	0.165
x30	0.1730	0.041	4.267	0.000	0.094	0.252
x31	0.1623	0.036	4.501	0.000	0.092	0.233
x32	0.1621	0.034	4.741	0.000	0.095	0.229
x33	−0.0332	0.022	−1.491	0.136	−0.077	0.010
x34	0.0384	0.039	0.977	0.329	−0.039	0.115
x35	0.0096	0.045	0.215	0.830	−0.078	0.097
x36	0.1829	0.067	2.719	0.007	0.051	0.315

Omnibus:	2665.268	Durbin-Watson:	2.006
Prob(Omnibus):	0.000	Jarque-Bera (JB):	12940.010
Skew:	−1.760	Prob(JB):	0.00
Kurtosis:	8.594	Cond. No.	1.63e+16

OLS Regression Results

Dep. Variable:	Rating	R-squared:	0.032
Model:	OLS	Adj. R-squared:	0.028
Method:	Least Squares	F-statistic:	6.770
Date:	Tue, 02 May 2023	Prob (F-statistic):	3.28e-31
Time:	17:57:12	Log-Likelihood:	-5345.3
No. Observations:	7108	AIC:	1.076e+04
Df Residuals:	7072	BIC:	1.101e+04
Df Model:	35		
Covariance Type:	nonrobust		

- Most p value <0.05
- Low R squared
- MSE = 0.269
- Too many variables might lead to overfitting
- Solution: regularization

Data Modeling

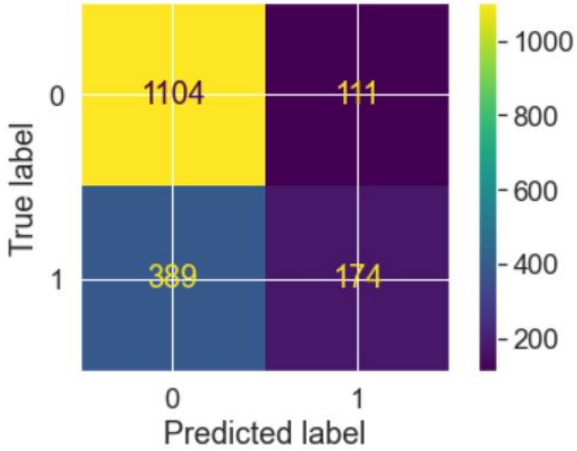
Regularization

- Tried Lasso, Ridge and Elastic Net

```
mean_squared_error score for Ridge: 0.266
mean_squared_error score for Lasso: 0.275
mean_squared_error score for Elastic Net: 0.266
```

Classification: explore non-linear relationship

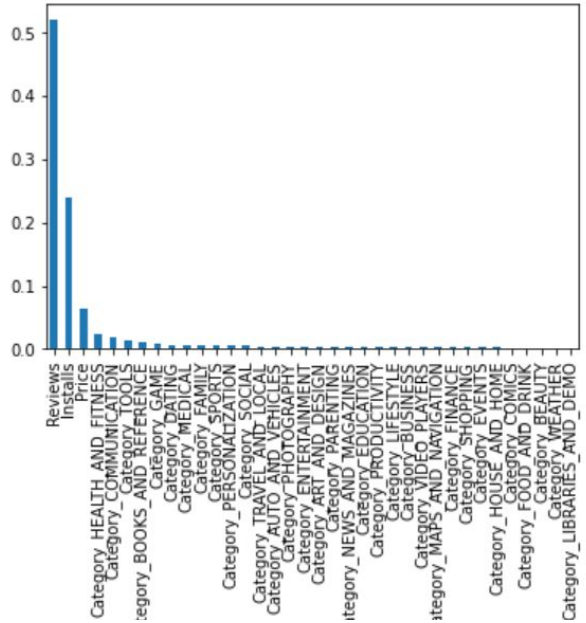
- Set rating greater than 4.5 to be 1 (high rating) and less than 4.5 to be 0 (low rating)
- Tried logistic regression, with accuracy score 0.729
- Did not improve generalization performance
- Tried ensemble later

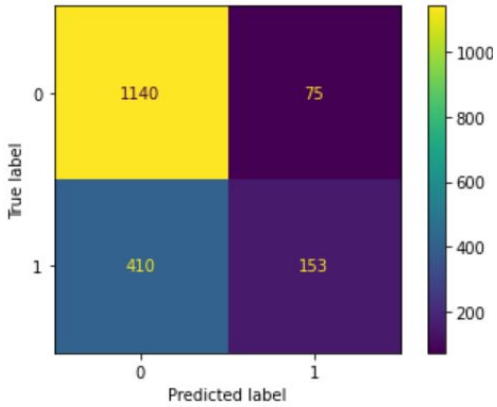


Data Modeling

Random Forest Classification

- Use Grid Search & CV to find the best hyperparameters
- Random forest Accuracy: 0.73
- Precision: 0.67. Low Recall: 0.27
- Label 0: low rating. Label 1: high rating
- Fairly small false negative cases, where we predict the app to have high rating when it does not.
- Do not want the business to waste resources on those apps.
- The most important features are: number of reviews, number of installs.
- We will analyze the most important variable ‘Review’ in detail later.





Sentiment analysis

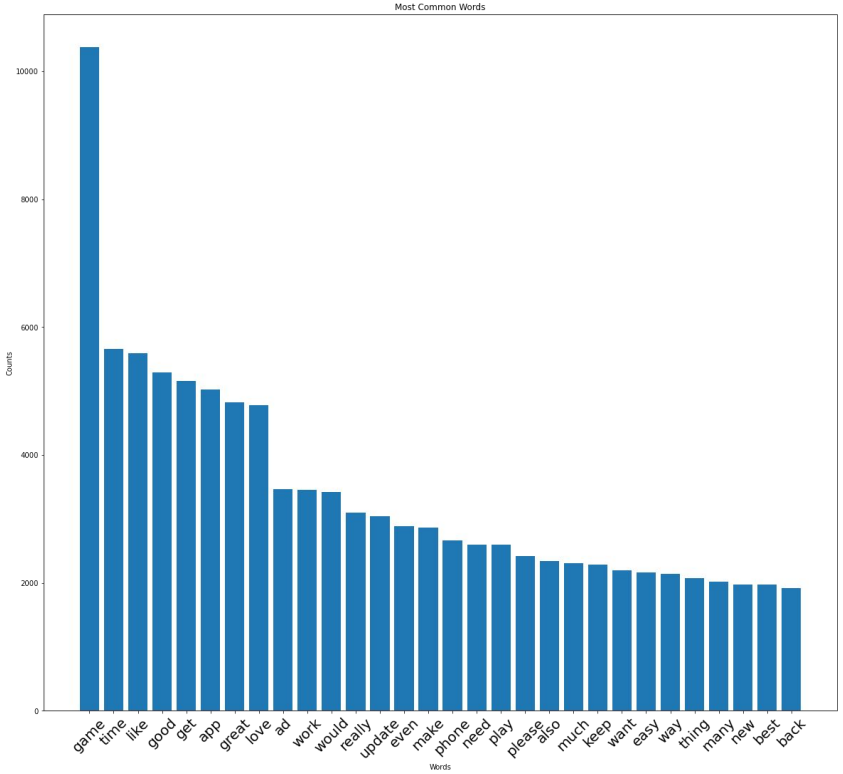
- Our dataframe contains first 'most relevant' 100 reviews for each app.
- Assign 0 as Positive sentiment; 1 as Negative; 2 as Neutral.
- For each string in Translated_Review, remove stopwords, unnecessary characters, and perform lemmatization.
- Generate bag of words–the frequency of each word to be used as a feature

	App	Translated_Review	Sentiment	Sentiment_Polarity	Sentiment_Subjectivity
0	10 Best Foods for You	I like eat delicious food. That's I'm cooking ...	Positive	1.00	0.533333
1	10 Best Foods for You	This help eating healthy exercise regular basis	Positive	0.25	0.288462
2	10 Best Foods for You	NaN	NaN	NaN	NaN
3	10 Best Foods for You	Works great especially going grocery store	Positive	0.40	0.875000
4	10 Best Foods for You	Best idea us	Positive	1.00	0.300000

Sentiment analysis

- Word cloud of the most common words in negative reviews (left) and positive reviews (right)

Sentiment analysis



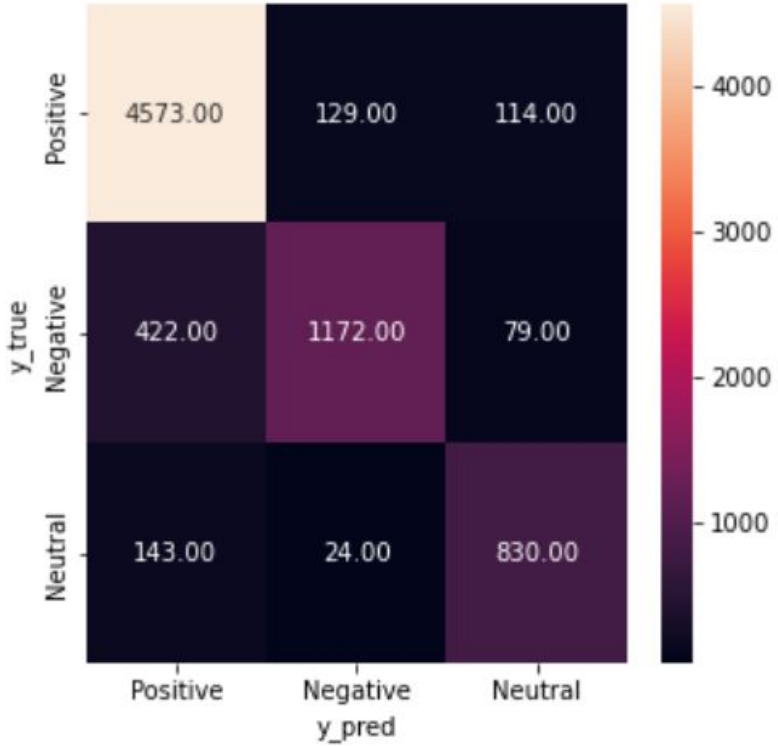
- Bar chart of the most common words in all reviews

Sentiment analysis

Random Forest Classification modeling

- Split the data into training and test data
- Use processed review as features, sentiment as labels
- High Accuracy: 88.47%

	precision	recall	f1-score	support
0	0.90	0.95	0.92	4812
1	0.89	0.73	0.80	1640
2	0.82	0.84	0.83	1034
accuracy			0.88	7486
macro avg	0.87	0.84	0.85	7486
weighted avg	0.88	0.88	0.88	7486

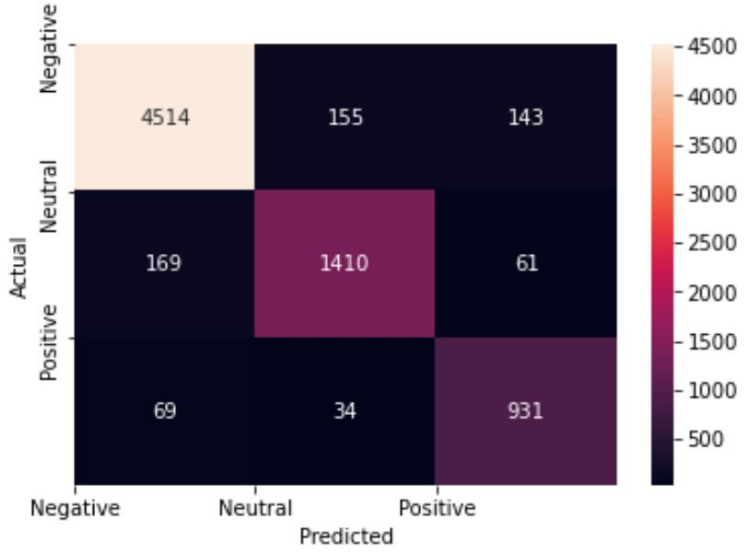


Sentiment analysis

Logistic regression modeling

- Higher accuracy: 96.81%
- Also, better precision, recall and F1-score than random forest classification

	precision	recall	f1-score	support
0	0.95	0.94	0.94	4812
1	0.88	0.86	0.87	1640
2	0.82	0.90	0.86	1034
accuracy			0.92	7486
macro avg	0.88	0.90	0.89	7486
weighted avg	0.92	0.92	0.92	7486



Insights and Conclusion

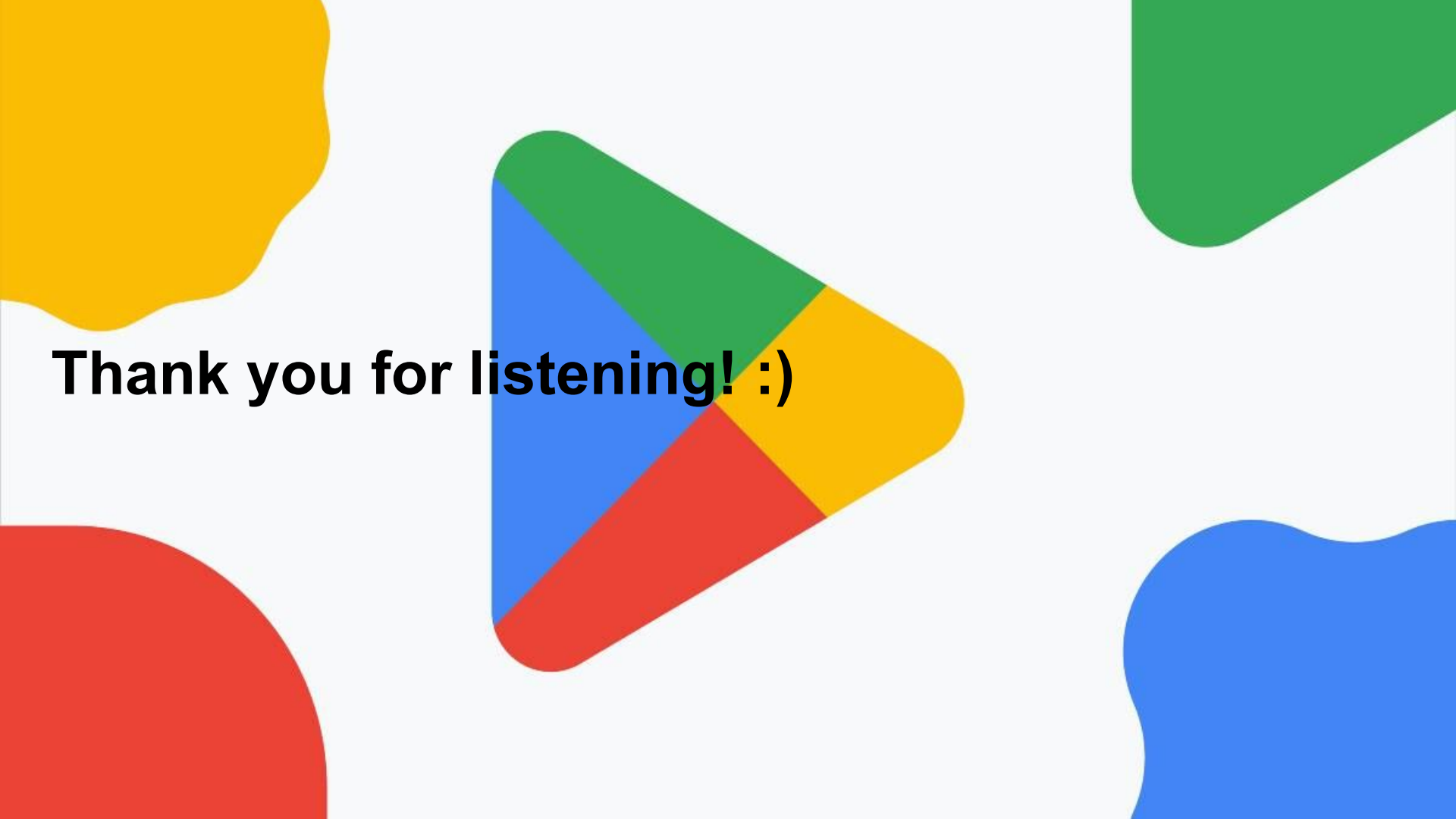
- Overall, in both regression and classification models, reviews & installation amount play significant roles in the prediction of APPs rating.
- It's important to clarify the linear or non-linear relationship between features when modeling.
- However, there exist some bias for app rating without considering the uninstalling rate:
  For instance: people who has negative attitude on APP might directly delete the app, and such unrecorded outcomes may affect the APPs rating score prediction.
- For Improvement: inquire more detailed information about apps, can we focus on specific category of apps to make analysis, so that produce more targeted and effective insights for specific field.
- For business, this could be beneficial for encouraging higher active rate of software development especially in certain category (Gaming and Family), increasing user engagements as well as Google's business development.

Insights and Conclusion

- By analyzing user reviews, developers can gain insights into the overall sentiment towards their app, as well as specific features or aspects of the app that users like or dislike.
- Identifying areas for improvement: (Negative sentiment)

  Developers can identify aspect of the app that are causing frustration or dissatisfaction for users. Those information can be used to make changes that will improve the user experience and increase user satisfaction.
- Understanding user behavior: (Positive sentiment)

  Sentiment analysis can also provide insights into user behavior and preferences. For example, if users consistently mention a particular feature in their positive reviews, developers can focus on improving and promoting that feature to increase user satisfaction.


Thank you for listening! :)