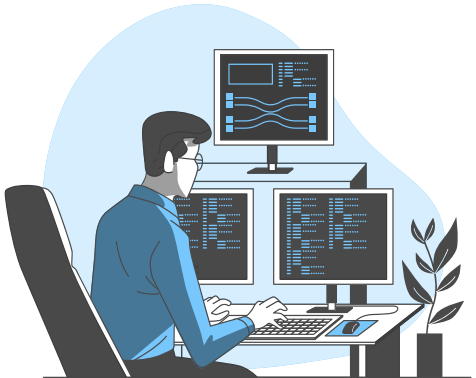
SDAIA BOOTCAMP T5 PROJECT

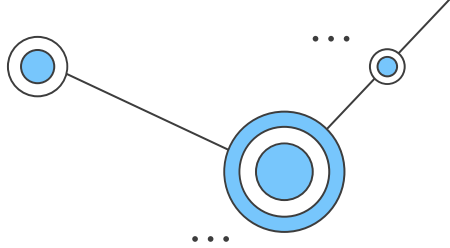
Hessa Ali Alhamad

CAN WE PREDICT THE DEMAND OF APPS BASED ON GOOGLE PLAY STORE DATA?

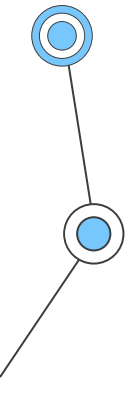


CAN WE PREDICT AN APP RATING?

INTRODUCTION

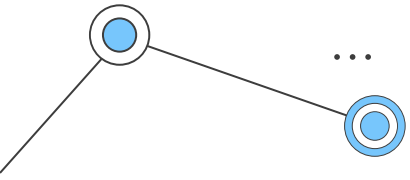


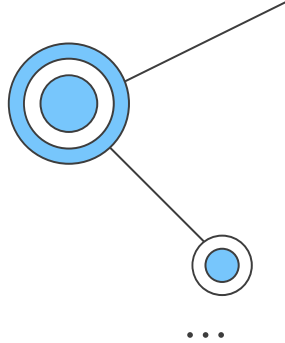
With technology dominating the modern world; the app industry has become a high thriving. New apps are coming to the market every while. Indeed, app usage is still growing at a steady rate, but some are higher than others. There are different demands for different apps based on several features. Data science potentials can be utilized to drive app-making businesses and app developers to the right road.



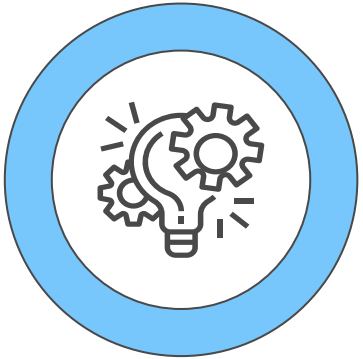
DATASET DESCRIPTION

Google Play Store is a big digital distribution service that provides apps supported by Android-certified devices and Chrome OS. We found a dataset on Kaggle that contains data of 10k Play Store apps for analyzing the Android market. The dataset has 13 columns -which will be shown in the following subsection- and 10842 rows. It is aiming to use these apps' statistics to predict which apps are more likely to be installed or get a high rate.





	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	10,000+	Free	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	500,000+	Free	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	5,000,000+	Free	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	50,000,000+	Free	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	100,000+	Free	0	Everyone	Art & Design;Creativity	June 20, 2018	1.1	4.4 and up



PURPOSE OF THE PROJECT

There are two principle goals for this project. The first goal is to classify apps based on their Installs and Rating, taking into consideration other features like Category, Type, Price and Content Rating. We're aiming to run many experiments with different classifier models and many trails to discover their effect on the accuracy scores. The second goal of this project is to predict Rating based on set of app features like Category, Reviews, Size, Installs, Type, Price and Content Rating. We're aiming to build a regression model that can predict the app rating.

DATA CLEANING

01

Remove Duplicates

We found 1181 duplicated in the dataset

02

Drop Irrelevant Columns

('App', 'Genres', 'Last Updated', 'Current Ver', 'Android Ver')

03

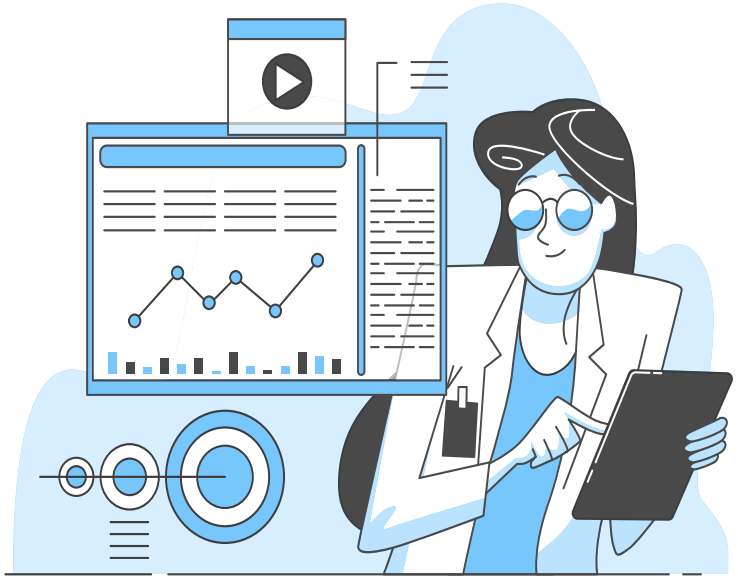
Handle Missing Values

We found 1465 NaN values, we filled most of them with mean value.

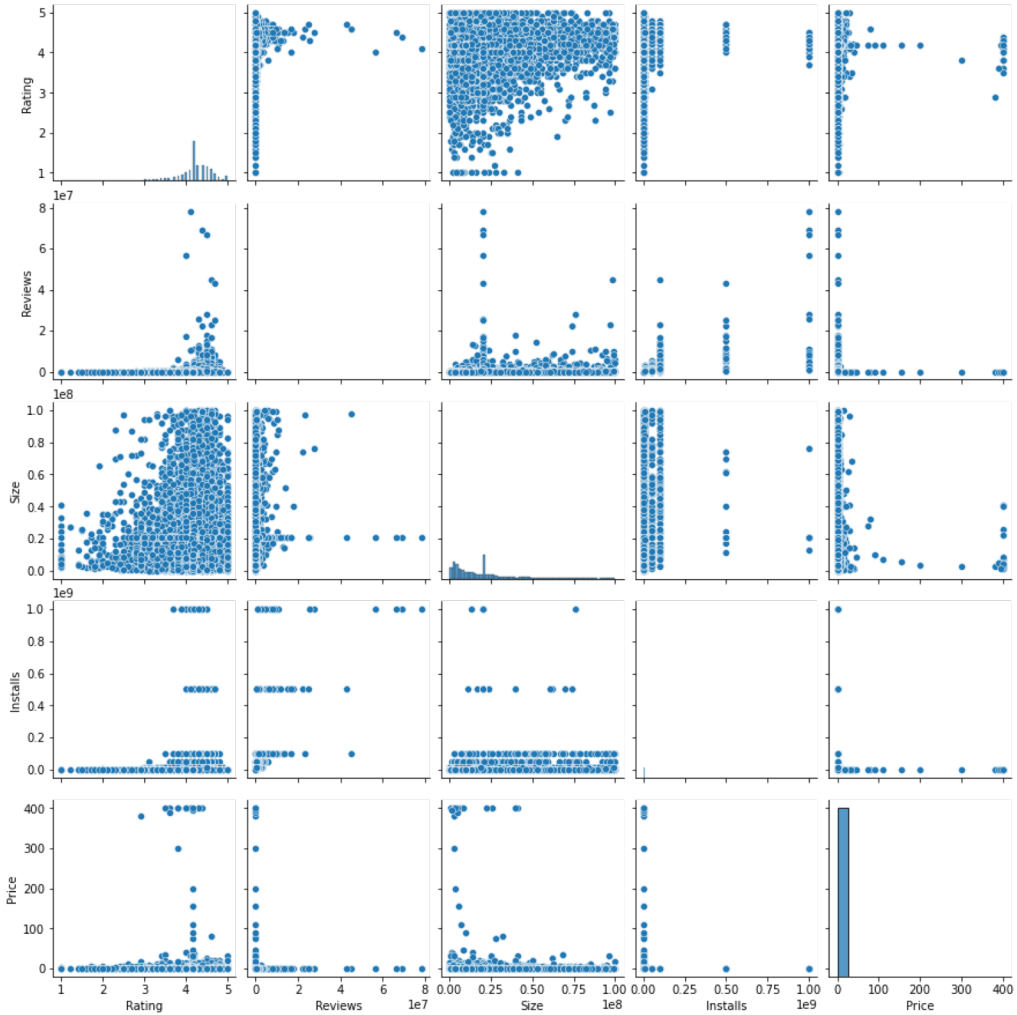
04

Change Dtype

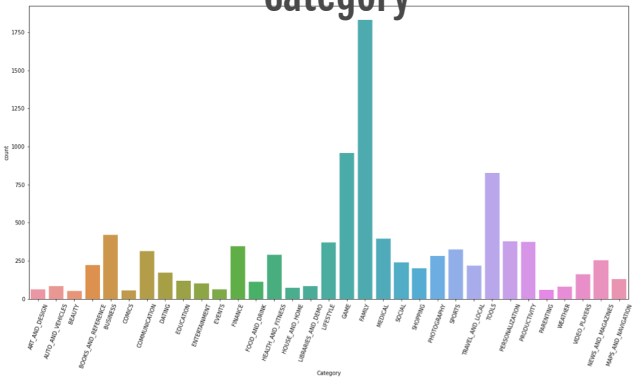
We changed some 'object' dtypes to numeric dtypes. We had to do some replacement and filling to feature values first.



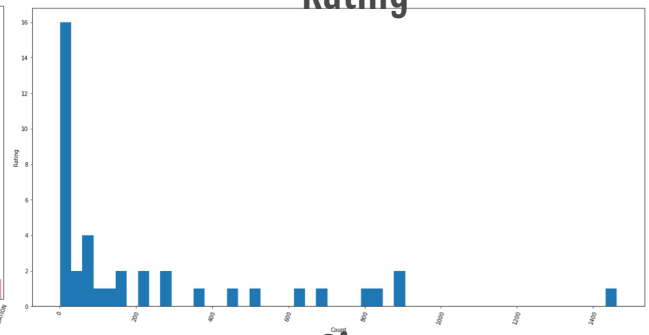
VISUALIZATION



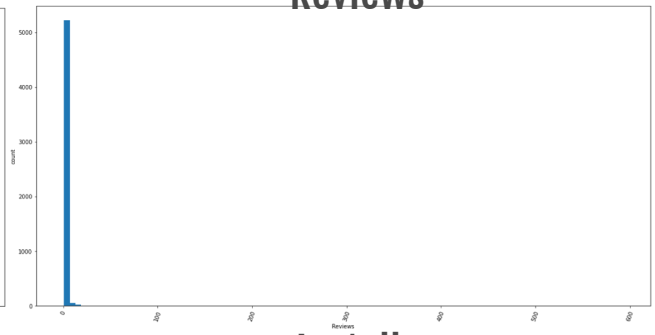
Category



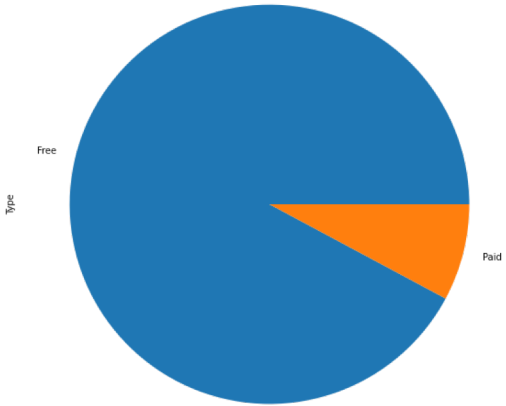
Rating



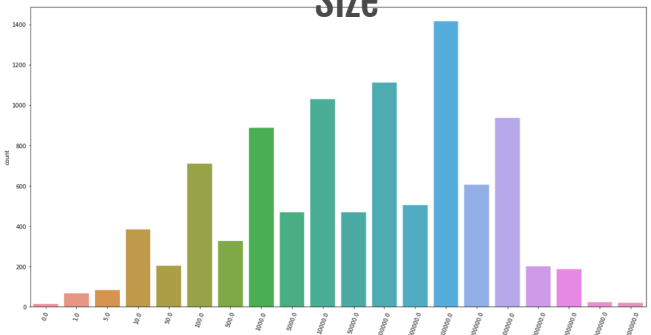
Reviews



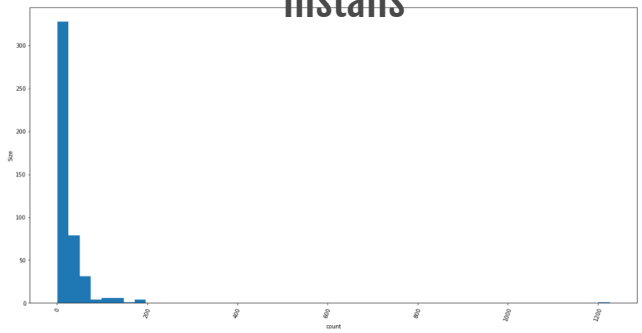
Type



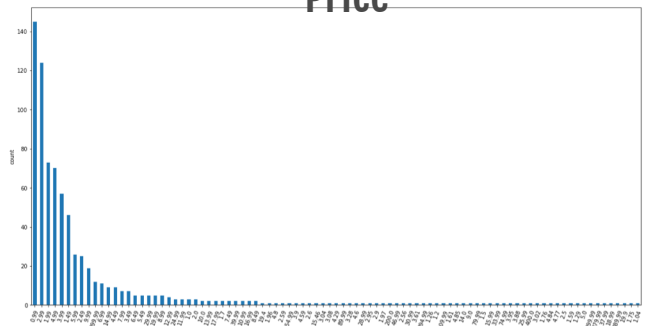
Size



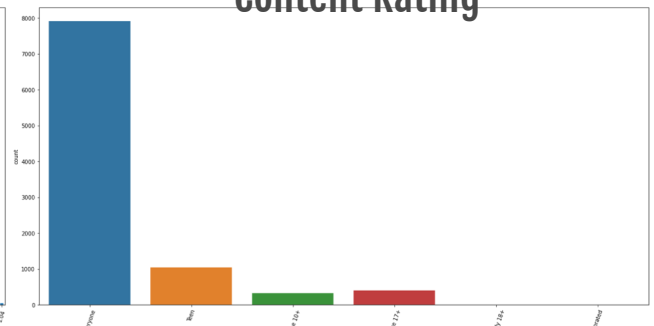
Installs



Price



Content Rating



FEATUARE ENGINEERING

We did add a new column called (App_Demand) engineered from the 'Installs' and 'Rating' features. The column have 7 categories ('very_high_demand' , 'high_demand', 'on_demand', 'moderate_demand', 'low_demand', 'very_low_demand', 'no_demand') which represent the demand on a certain app based on its installs and rating.

FEATUARE ENGINEERING

01 Calculations Related to New Column

We computed maximum, average and minimum 'Installs' values, and average and minimum 'Rating' values using max(), mean(), and min() methods respectively.

02 Additional Variables

We defined two variables, (lowrRateIN) which denotes the average 'Installs' value between the minimum and average values. And (highRateIN) which denotes the average 'Installs' value between the maximum and average values.

FEATUARE ENGINEERING

03 Inserting the New Column

The new column called (App_Demand) which has 7 categories

First, the 'very_high_demand' apps are the app with the installs number between maximum and highRateIN values.

Second, the 'high_demand' apps are the app with the installs number between highRateIN and average value and rating value above average value.

Third, the 'on_demand' apps are the app with the installs number between highRateIN and average values and rating value below average value.

Fourth, the 'moderate_demand' apps are the app with the installs number between average and lowrRateIN values.

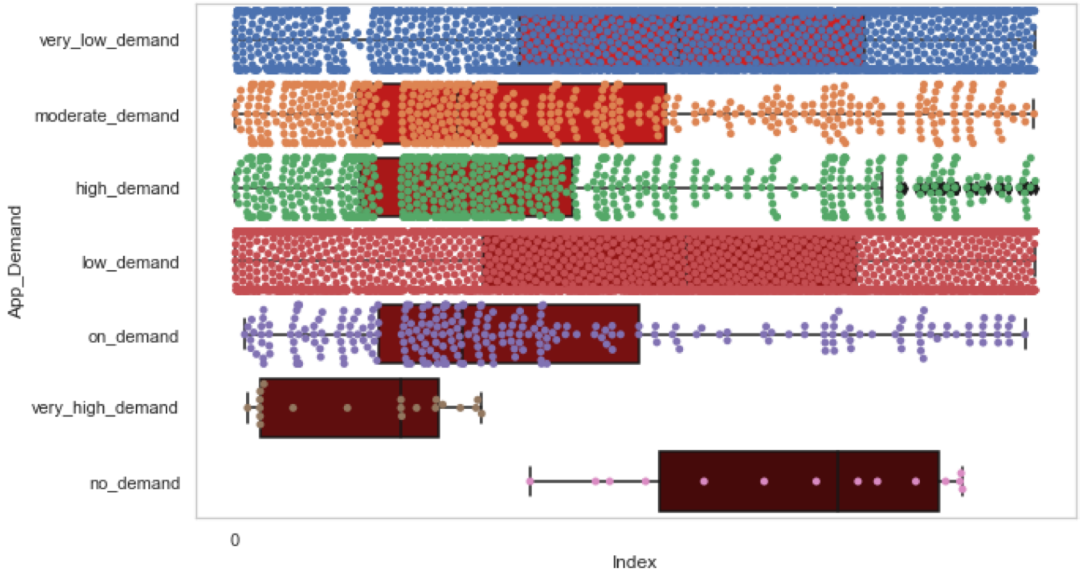
Fifth, 'low_demand' apps are the app with the installs number between lowrRateIN and minimum values and rating value above average value.

Sixth, 'very_low_demand' apps are the app with the installs number between lowrRateIN and minimum values and rating value below average value.

The last category, 'no_demand' apps are the app with the installs number equal to minimum values.

FEATUARE ENGINEERING

	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	App_Demand
0	ART_AND_DESIGN	4.1	159.0	19000000.0	10000.0	Free	0.0	Everyone	very_low_demand
1	ART_AND_DESIGN	3.9	967.0	14000000.0	500000.0	Free	0.0	Everyone	very_low_demand
2	ART_AND_DESIGN	4.7	87510.0	8700000.0	5000000.0	Free	0.0	Everyone	moderate_demand
3	ART_AND_DESIGN	4.5	215644.0	25000000.0	50000000.0	Free	0.0	Teen	high_demand
4	ART_AND_DESIGN	4.3	967.0	2800000.0	100000.0	Free	0.0	Everyone	low_demand



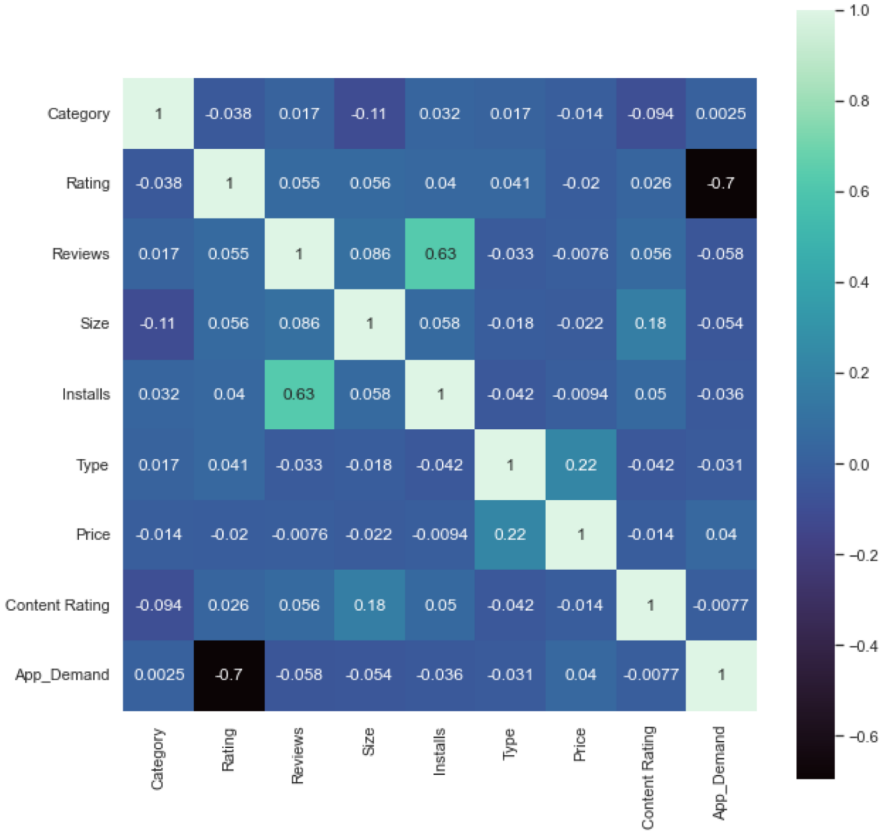
DATA TRANSFORMATION

We encoded categorical labels into numerical values.
We used the LabelEncoder from sklearn.preprocessing library.

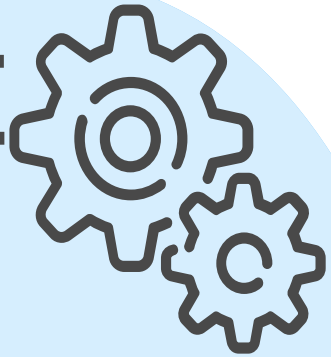
- Category
- Type
- Content Rating
- App_Demand

	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	App_Demand
0	0	4.1	159.0	19000000.0	10000.0	0	0.0	1	6
1	0	3.9	967.0	14000000.0	500000.0	0	0.0	1	6
2	0	4.7	87510.0	8700000.0	5000000.0	0	0.0	1	2

CORRELATION MATRIX



BUILDING THE MODELS



First, we developed K-Nearest Neighbor and Random Forrest Classifier models for classifying the apps into categories based on their demand.

Second, we plan to develop a Random Forrest Regression model to predict the app rating.

CLASSIFICATION GOAL

Set Features and Target

('Category', 'Type', 'Price', 'Content Rating') features for X, The new feature (App_Demand) for y

Splitting the Dataset

We set the test_size parameter 0.30

Feature Scaling

Using MinMaxScaler() to be in range between zero and one

Training Models

W perform set of experiments to test the results

Evaluating Models

We evaluated each experiment

K-Nearest Neighbor Classifier

	Trial 1	Trial 2
Overview	We only tuned for best value of k using cross validation	Resampled the dataset using SMOTE and find k and build the mode on it
Training set accuracy	0.531508875739645	0.3874980003199488
Testing set accuracy	0.525879917184265	0.3053830227743271
Confusion Matrix	[[33 289 0 0 0 0 0] [38 1491 0 0 0 0 2] [9 173 0 0 0 0 0] [0 4 0 0 0 0 0] [7 77 0 0 0 0 0] [1 5 0 0 0 0 0] [35 734 0 0 0 0 0]]	[[196 49 44 1 9 20 3] [445 623 176 170 16 47 54] [78 52 31 0 7 11 3] [0 0 0 3 0 0 1] [47 19 9 0 2 7 0] [4 1 0 0 0 1 0] [287 282 95 42 11 23 29]]

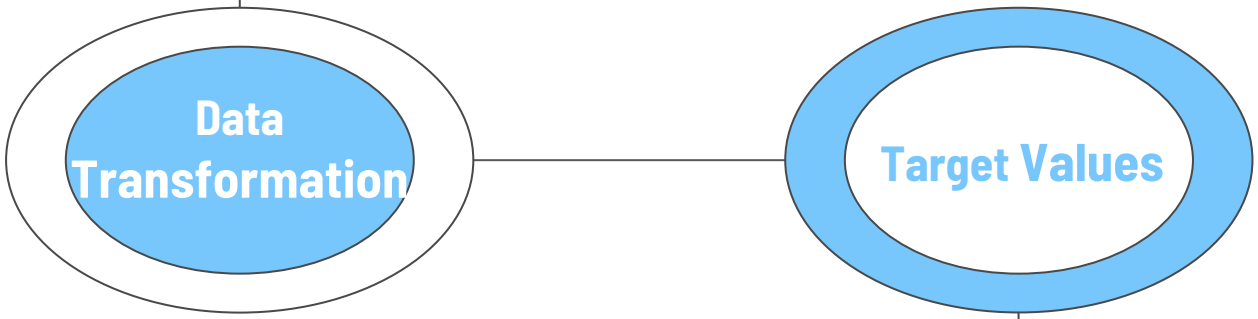
Random Forrest Classifier

	Trial 1	Trial 2	Trial 3
Overview	An initial experiment	We used (class weight) parameter and set it to 'balanced'	Resampled the dataset using SMOTE and find k and build the mode on it
Training set accuracy	0.5455621301775148	0.28816568047337277	0.28943369060950247
Testing set accuracy	0.5158730158730159	0.25086266390614215	0.28778467908902694
Confusion Matrix	[[39 266 1 0 0 0 16] [51 1411 1 0 0 0 68] [14 155 0 0 0 0 13] [0 4 0 0 0 0 0] [13 66 0 0 0 0 5] [2 4 0 0 0 0 0] [40 683 1 0 0 0 45]]	[[139 8 51 4 51 30 39] [217 321 179 125 191 62 436] [40 9 24 4 32 27 46] [0 3 0 0 0 0 1] [25 3 9 1 20 10 16] [2 0 0 0 1 3 0] [129 96 80 52 138 54 220]]	[[135 12 53 2 51 34 35] [219 429 181 18 183 72 429] [43 14 24 0 34 22 45] [0 1 0 3 0 0 0] [23 3 9 0 24 9 16] [1 0 1 0 1 3 0] [129 141 79 4 143 57 216]]

CLASSIFICATION GOAL

Before building the regression model

We log-transformed the 'Installs' column



We dropped a target value that was in a single data point

REGRESSION GOAL

Set Features and Target

All the remaining columns exept 'Rating','App_Demand' X,
The feature ('Rating') for y

Splitting the Dataset

We used different test size split to see their results (.15, .30, .50, .75)

Feature Scaling

Using MinMaxScaler() to be in range between 0 and 1

Training Models

W trained the model using Random Forest Regressor

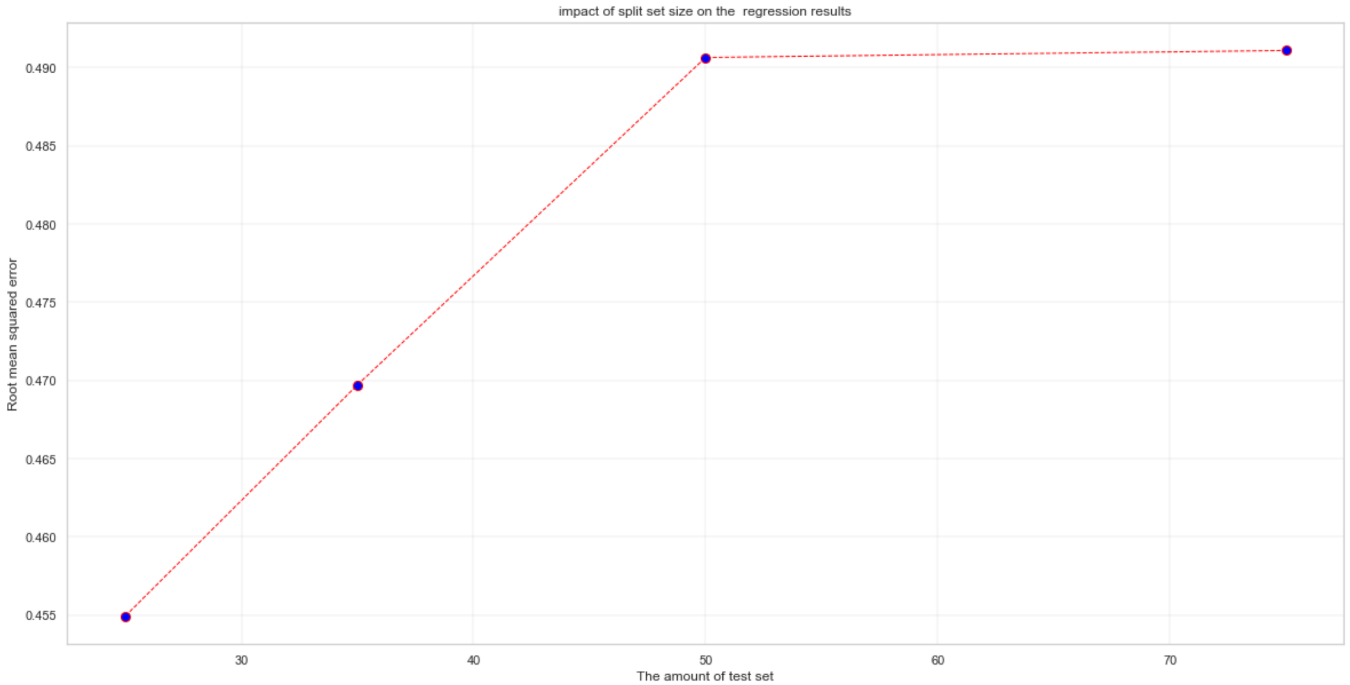
Evaluating Models

We evaluated the model with RMSE

Random Forrest Classifier

	Trial 1	Trial 2	Trial 3	Trial 4
Overview	Training set > Testing set (15)	Training set > Testing set (30)	Training set == Testing set (50)	Training set < Testing set (75)
RMSE	0.45493198981234695	0.46971027534528054	0.49064546580455526	0.49109259386519327

CORRELATION MATRIX



CONCLUSIONS

Depending on this dataset we could not predict the demand of apps based on their category, type, price and content rating. We attribute that to some reasons:

- The dataset is highly imbalanced.
- There were no strong relationships among features.

Regarding the predicting of the rating, we –somehow- reach some reasonable results. For further, we suggest building different regression algorithms to find the best result.

Thanks!

Do you have any questions?