

Prediction of Google Play Store Application Ratings

A PROJECT REPORT

Submitted by

TRISHA PATEL

200103042189

BACHELOR OF ENGINEERING

in

Computer Engineering



College of Technology

Silver Oak College of Engineering & Technology



Silver Oak University, Ahmedabad

MAY, 2024



Silver Oak College of Engineering & Technology

Opp. Bhagwat Vidhyapith, S.G. Highway, Ahmedabad-382481

CERTIFICATE

This is to certify that the Project report submitted along with the Internship entitled **Prediction of Google Play Store Application Ratings** has been carried out by **Trisha Patel** under my guidance in partial fulfillment for the Bachelor of Engineering in Computer Engineering, 8th Semester of Silver Oak University, Ahmedabad during the academic year 2023-24.

A/Prof. Gaurav Tiwari
Internal Guide

DR. Satvik Khara
Head of the Department



301-305, 3rd Floor, Surabhi Complex, Nr. Municipal Market, C.G.Road, Navrangpura, Ahmedabad-380009

Date: 08/04/2024

TO WHOM IT MAY CONCERN

This is to certify that **Ms Trisha Patel**, a student with Enrollment no.200103042189, Sem 8th, Department Computer of Silver Oak College of Engineering & Technology has successfully completed her internship in the field of python from 28th July 2022 to 28th June 2024 under the guidance of Mr. Rahul Kirpekar.

Her internship activities include:

- Learning Sessions.
- Code review and Testing.
- Documentation and Commenting.
- Bug Fixing.
- Building Projects.

During the period of her internship program with us, she had been exposed to different processes and was found diligent, hardworking and inquisitive.

We wish him every success in her life and career.

Sincerely

A handwritten signature in blue ink is written over a circular purple stamp. The stamp contains the text "GROWNITED PRIVATE LIMITED" around the perimeter and "GROWNITED" in the center.

Rahul Kirpekar
(Authorised Signature)



Silver Oak College of Engineering & Technology

Opp. Bhagwat Vidhyapith, S.G. Highway, Ahmedabad-382481

DECLARATION

We hereby declare that the Internship report submitted along with the Internship entitled **Prediction of Google Play Store Application Ratings** submitted in partial fulfillment for the Bachelor of Engineering in Computer Engineering to Silver Oak University, Ahmedabad, is a bonafide record of original project work carried out by me at Grownited Private Limited under the supervision of Rahul Kirpekar and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student

Trisha Patel

Sign of Student

Acknowledgment

I would like to extend my heartfelt thanks with a deep sense of gratitude and respect to all those who have provided us immense help and guidance during my project. I would like to express my sincere thanks to my faculty guide A/Prof. Gaurav Tiwari for providing a vision about the system and for giving me an opportunity to undertake such a great challenging and innovative work. I am grateful for the guidance, encouragement, understanding and insightful support given in the development process. I would like to extend my gratitude to Dr. Satvik Khara Head of Computer Engineering Department, Silver Oak College of Engineering and Technology, Ahmedabad, for his continuous encouragement and motivation.

Yours Sincerely,

Trisha Patel

200103042189

Abstract

Analyzing app data from the Play Store helps app makers and the Android ecosystem. It gives developers useful insights into what users like, what's popular, and how well apps are doing. With this info, developers can improve their apps and make sure they're seen and liked by users.

This data isn't just helpful for developers; it's good for users too. It helps them pick the right apps for them, making sure they're happy with what they choose. And for developers, having this data is like having a map. It shows them where to go next to meet what users want and keep them happy. So, using data from the Play Store isn't just about numbers; it's about making better apps and keeping everyone satisfied.

LIST OF FIGURES

List Of Figures	Pg No.
1. System Flow Diagram	03
2. Block Diagram	03
3. Number of null and not null values in dataset	04
4. Total number of instances	04
5. Integer encoding of category and genres column	05
6. One-hot encoding of category column	05
7. Datatype for each attribute	06
8. Correlation matrix	07
9. Frequency of rating column	08
10. Rating vs Size	08
11. Rating vs Category	09
12. One-hot encoding of category column	09
13. Reviews vs Rating	10
14. Price vs Rating	11
15. Rating vs size	11
16. Reviews vs installs	12
17. Analyzing iterations for model	17
18. Comparison of different models	19

LISTS OF TABLE

List of Tables	Pg No.
1. Description of attributes	02
2. Linear regression model result	14
3. Random forest regressor model result	14
4. Support vector regression (SVM) model result	15
5. Comparison using different activation function with models	16
6. MLPRegressor model (deep neural network) result	17

Table of Content

Acknowledgment.....	iv
Abstract.....	v
List of Figures.....	vi
List of Tables.....	vii
Table of Contents.....	viii
Chapter 1. Introduction	1
1.1 Problem Domain	1
1.2 Proposed Solution	1
1.3 Dataset Description	2
Chapter 2. System Design.....	3
2.1 System Flow.....	3
2.2 Block Diagram	3
Chapter 3. Data Preprocessing	4
3.1 Data cleaning	4
3.2 Data Transformation	5
Chapter 4. Data Visualization.....	7
4.1 Correlation matrix of different data	7
4.2 Comparison between Attributes	7
Chapter 5. Models	13
5.1 Linear Regression	13
5.2 Random Forest Regressor	14
5.3 Support Vector Regression	15
5.4 Neural Network.....	15
5.4.1 Hyper-Parameter Tuning	15
5.4.2 Result	17
Chapter 6. Comparison of models	19
Chapter 7. Conclusion	21
Chapter 8.References.....	22

Chapter 1

Introduction

1. Introduction

In the current generation, mobile applications play a very important role in everyone's life. Substantial challenges from all directions cause much confusion for designers to create something new and effective. Application clients can make decisions after analysis regarding which type of application a person can make and what type of features are necessary to make it successful. The Google Play Store is observed as the largest application platform currently holding approximately 3.04 million applications, as reported by one article.

1.1 Problem Domain

In today's competitive world, it is very hard to create something attractive and user friendly. When a client asks to make an application for a particular domain, the first phase relates to a very close analysis of the market. Success of a developed application depends on various aspects. An application rating provides a visual representation of a user's likeability towards the application, and as such, what improvements are required to attain a higher rating.

Many applications are available on Google Play Store, but one can only see a portion of all applications based on the geographical location. Google is known to have many hidden caveats apart from the location. Moreover, Google uses its own recommendations and user behavior learning strategies, so all users will see different apps when they visit the Google Play Store.

1.2 Proposed Solution

Rating prediction can be done with the help of different categories such as, number of installations, size, category type (free/paid), genres, and the number of application reviews from customers. As a solution to the problem at hand, different models can be built to predict the application ratings and compare them to get optimal results. The first phase is all about cleaning and preprocessing the given dataset to get the best results as the dataset had many uncleaned data such as null or missing values.

Python codes were written and executed through the Google Colab, a free cloud service. The CSV file, googleplaystore.csv was utilized. Application ratings were predicted through different regression models and a comparison of the results were made.

1.3 Dataset Description

The dataset contains 13 attributes and 10841 instances with some missing and NaN values. Two types of data were available within the dataset; Qualitative (for example, type of application free/paid is binary data) and second is Quantitative (for example, genres of applications are a discrete type of data).

Table 1 Description of attributes

Numbers	Attributes	Description
1	App	Name of the application
2	Category	Category type of application
3	Rating	Overall user rating for any application
4	Reviews	Number of user reviews per application
5	Size	Size of the application
6	Installs	Number of users installs the application
7	Type	Free/Paid
8	Price	Price of the application (zero or any)
9	Content Rating	targeted age group - everyone/Adults/ Teenagers
10	Genres	An application belongs to multiple genres
11	Last Updated	Date when the application was last updated
12	Current Version	Latest version of the application
13	Android Version	Minimum required android version

Chapter 2 System Design

2. System Design

2.1 System Flow

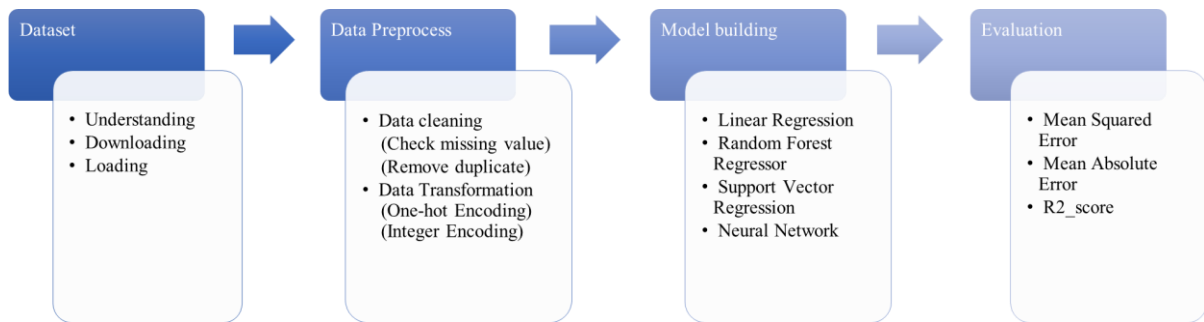


Figure 1 System Flow

2.2 Block Diagram

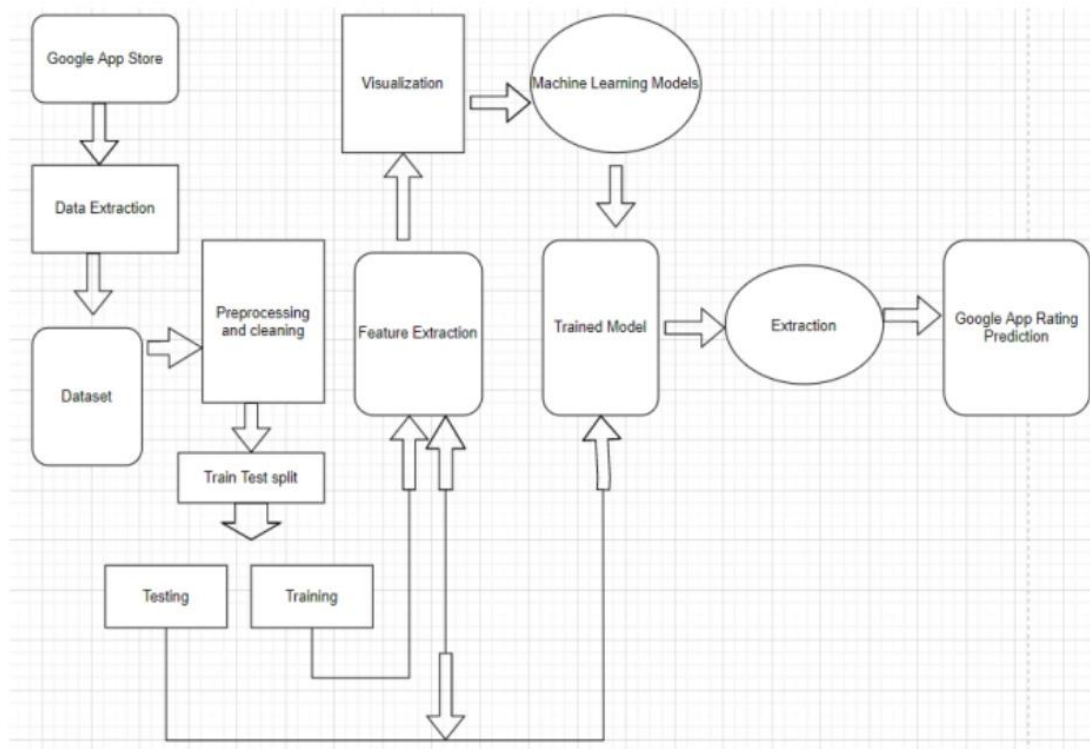


Figure 2 Block Diagram

Chapter 3

Data Preprocessing

3. Data Preprocessing

Data cleaning/preprocessing is the most important part of any machine learning process, as good quality data can give better predictions. Raw data can be taken as an input and make it suitable for a model.

3.1 Data cleaning

Data cleaning was used to identify and/or correct errors within the dataset. The given dataset had many missing values for the “Rating”, “Type”, “Content Rating”, “Current Ver” and “Android Ver” columns. The missing values needed to be removed and after removing the missing values, 9360 instances in total, were present.



 <code>df.isnull().sum()</code>	 <code>df.isnull().sum()</code>
 App 0	 App 0
Category 0	Category 0
Rating 1474	Rating 0
Reviews 0	Reviews 0
Size 0	Size 0
Installs 0	Installs 0
Type 1	Type 0
Price 0	Price 0
Content Rating 1	Content Rating 0
Genres 0	Genres 0
Last Updated 0	Last Updated 0
Current Ver 8	Current Ver 0
Android Ver 3	Android Ver 0
dtype: int64	dtype: int64

Figure 3 Number of null and not null values in dataset

 <code>df.info()</code>	 <code># after removing missing value we have 9360 instances</code> <code>df.info()</code>
 <code><class 'pandas.core.frame.DataFrame'></code> RangeIndex: 10841 entries, 0 to 10840 Data columns (total 13 columns):	 <code><class 'pandas.core.frame.DataFrame'></code> Int64Index: 9360 entries, 0 to 10840 Data columns (total 13 columns):

Figure 4 Total number of instances

Cleaning the size of the installation column was important as it had various types of number formats (for example, 2M or 1,4K), so they were all converted into integer numbers. Unnecessary columns of the dataset, which weren't required while performing models were

removed. The price column was converted into the float data type, as it has different prices like 0, 4.99\$, and so on. Next, the reviews column was converted into an integer datatype.

3.2 Data Transformation

For the category column, basically two methods were applied, one was integer encoding and second one was one-hot encoding. It is necessary to convert the categorical data (text values) into numeric values because it provides better prediction accuracy. Performing the integer encoding and transforming the category and genres columns into the “Category_c” and “Genres_c” columns, where both columns have numeric integer values was required.

Performing the one-hot encoding for categories giving different columns for each category (simply convert the rows with columns). One-hot encoding would not be used for genres column as it is a subset of category column, thus only integer encoding for genres column was considered.

In conclusion, four separate regression models were running, one including genres column, second excluding genres column, third integer encoding and forth one-hot encoding. In the next step, convert the application type (free/paid) into binary data (0/1), so we can use it easily in the model evaluation.

```
27] df.head()
```

	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Rating	Genres	Category_c	Genres_c
0	ART_AND_DESIGN	4.1	159	19000000.0	10000	0	0.0		0	Art & Design	0	0
1	ART_AND_DESIGN	3.9	967	14000000.0	500000	0	0.0		0	Art & Design;Pretend Play	0	1
2	ART_AND_DESIGN	4.7	87510	8700000.0	5000000	0	0.0		0	Art & Design	0	0
3	ART_AND_DESIGN	4.5	215644	25000000.0	50000000	0	0.0		1	Art & Design	0	0
4	ART_AND_DESIGN	4.3	967	2800000.0	100000	0	0.0		0	Art & Design;Creativity	0	2

Figure 5 Integer encoding of category and genres column

```
df2.head()
```

Category_ART_AND_DESIGN	Category_AUTO_AND_VEHICLES	Category_BEAUTY	Category_BOOKS_AND_REFERENCE	Category_BUSINESS	Category_COMICS	Category_COMMUNICATION
1	0	0	0	0	0	0
1	0	0	0	0	0	0
1	0	0	0	0	0	0
1	0	0	0	0	0	0
1	0	0	0	0	0	0

Figure 6 One-hot encoding of category column

Figure 5, shows the datatype of each attribute column after applying the data cleaning steps.

```

df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 9360 entries, 0 to 10840
Data columns (total 11 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Category             9360 non-null   object
1   Rating               9360 non-null   float64
2   Reviews              9360 non-null   int64
3   Size                 9360 non-null   float64
4   Installs              9360 non-null   int64
5   Type                 9360 non-null   int64
6   Price                9360 non-null   float64
7   Content Rating       9360 non-null   int64
8   Genres                9360 non-null   object
9   Category_c           9360 non-null   int64
10  Genres_c              9360 non-null   int64
dtypes: float64(3), int64(6), object(2)
memory usage: 877.5+ KB

```

Figure 7 Datatype for each attribute

Chapter 4 Data Visualization

4. Data Visualization

4.1 Correlation matrix of different data

This section contains correlations between Rating, Reviews, Size, Installs and Price data of the dataset. A moderate positive correlation of 0.64 exists between the number of reviews and number of downloads. This means that customers tend to download a given app more if it has been reviewed by a larger number of people. If you see the Rating data, Size of the application plays a very important role.

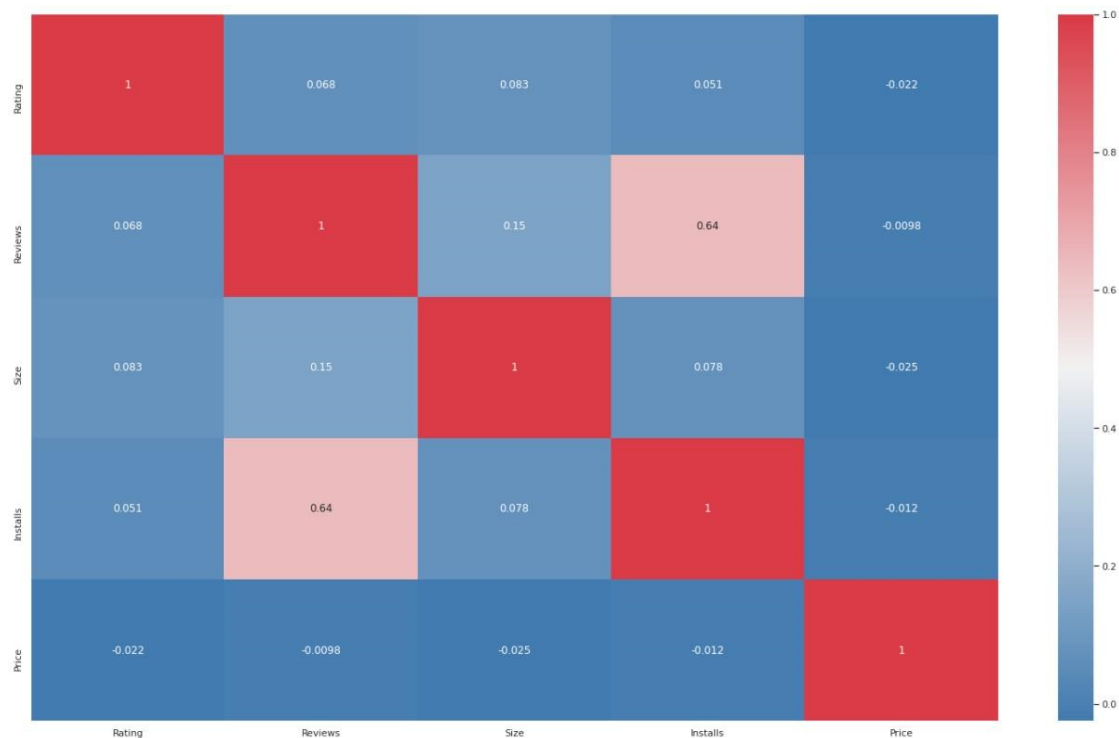


Figure 8 Correlation matrix

4.2 Comparison between Attributes

Figure 7, shows that most of the applications have ratings between 3 to 5, where the average rating of (active) applications on Google Play Store is 4.19. Figure 8, shows that the top-rated applications are optimally sized between 2MB to 40MB - neither too light nor too heavy in size.

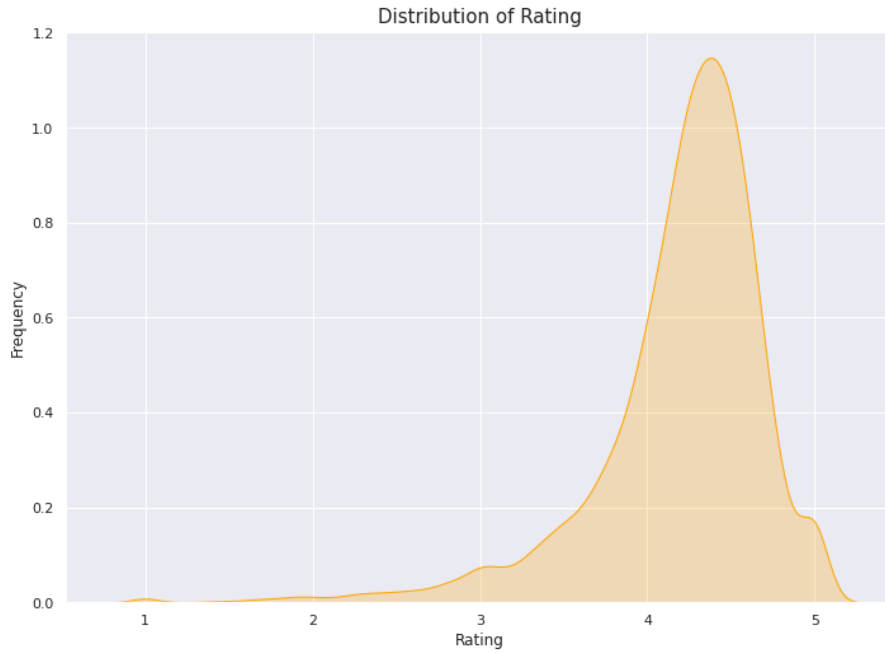


Figure 9 Frequency of rating column

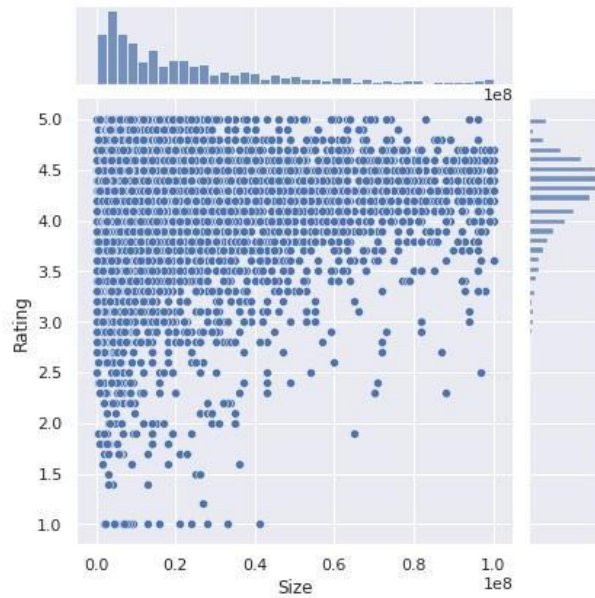


Figure 10 Rating vs Size

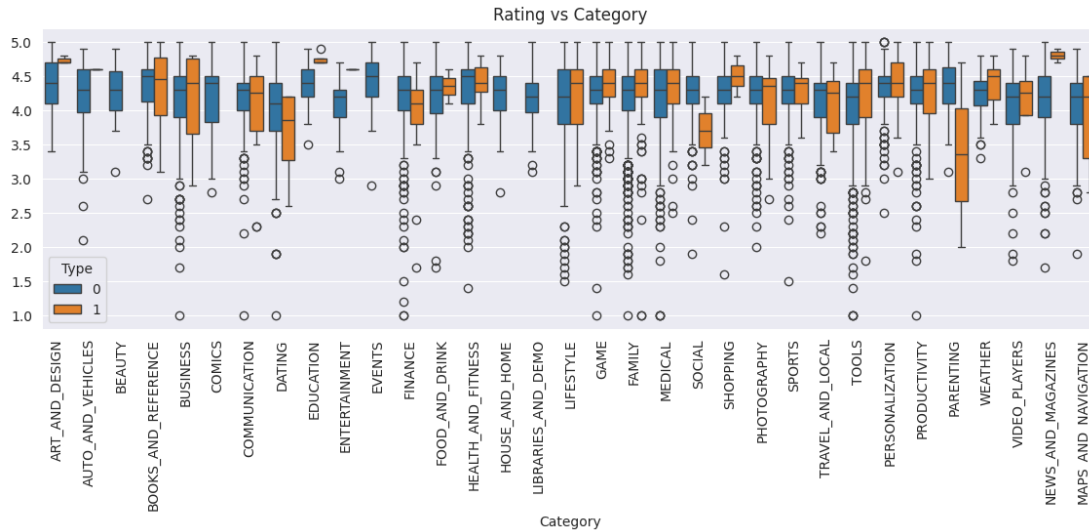


Figure 11 Rating vs Category

From this plot we can see that in most categories, paid apps have higher rating than free apps. In particular it is also interesting to notice that free apps have lots of outlier values compared to paid apps.

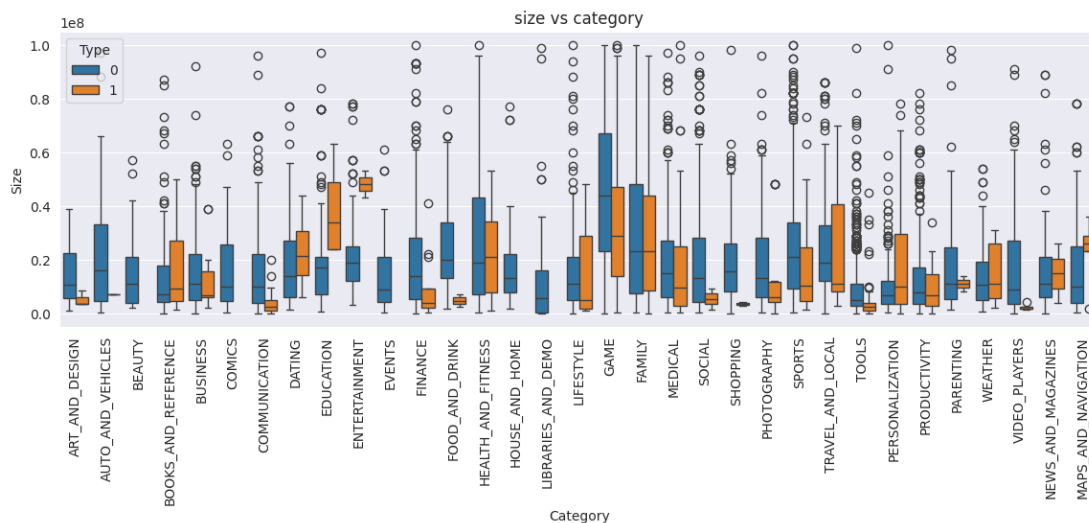
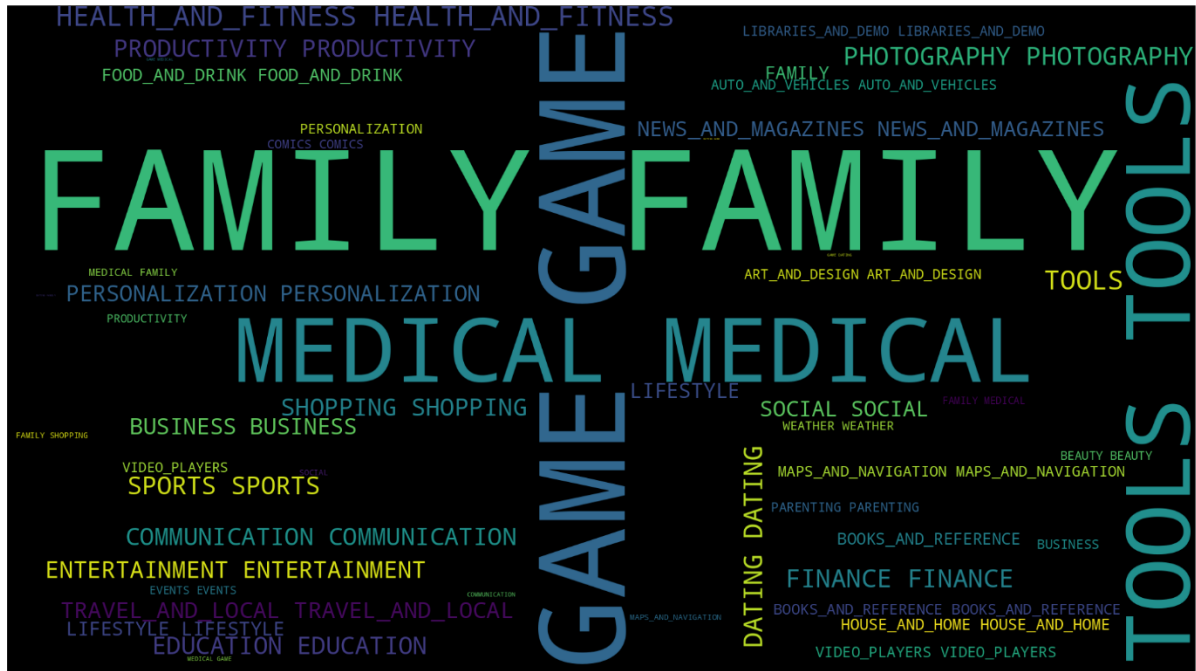


Figure 12 size vs category

We can see that the category where apps have a higher size are 'game', 'travel and local' (for paid apps only), education (for paid apps only) and family. In particular, free apps seem to have a higher size compared to paid apps for almost all categories.



We can see that the category where apps have a higher size are 'game', 'travel and local' (for paid apps only), education (for paid apps only) and family. In particular, free apps seems to have a higher size compared to paid apps for almost all categories.

Do apps with high rating have more reviews?

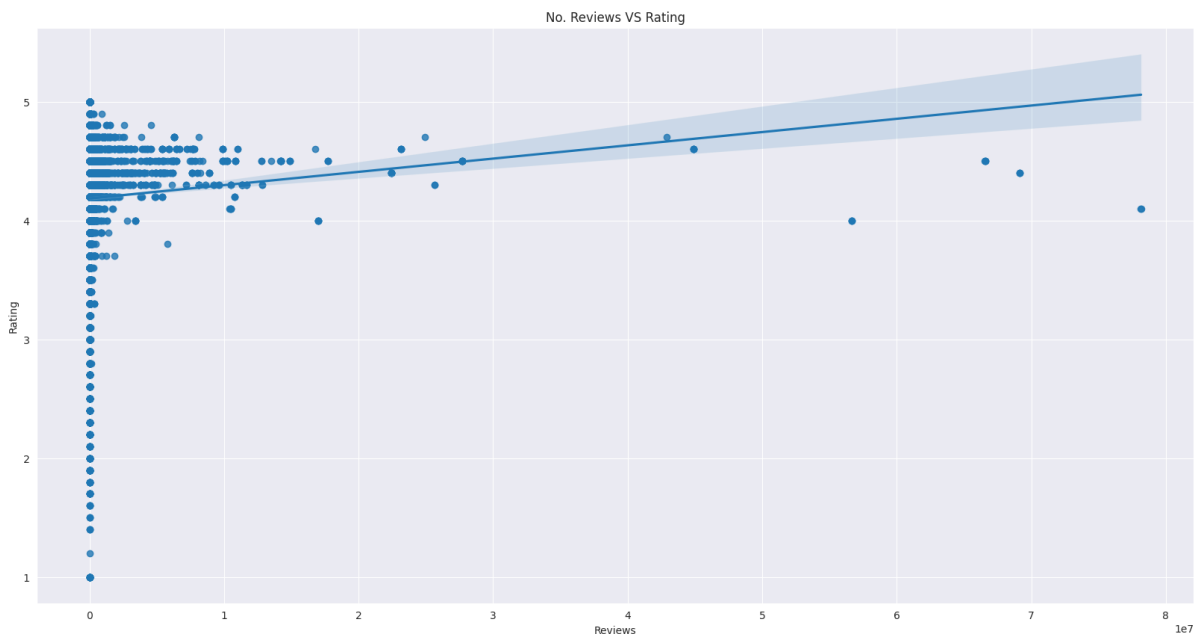


Figure 13 Reviews vs Rating

We can see a positive trend between rating and number of reviews: apps with more reviews tends to have higher rating.

Do expensive apps have higher rating?

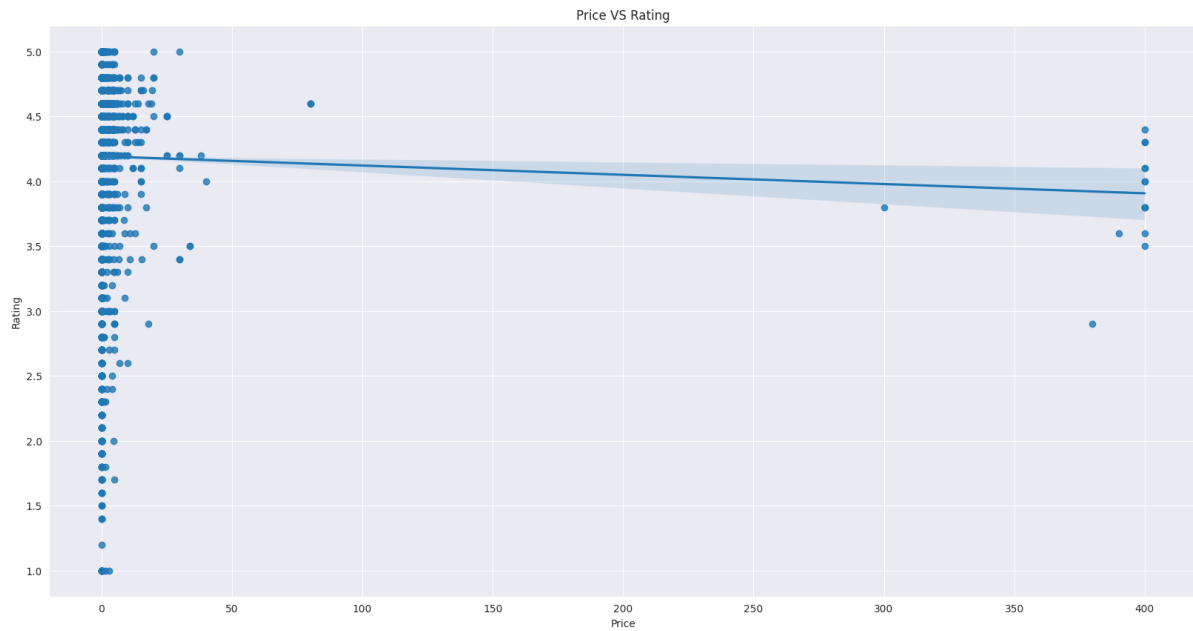


Figure 14 price vs Rating

From this plot we can see a slight positive trend between price and rating: apps with higher prices tends to be slightly higher rated.

Is there any relationship between app rating and size?

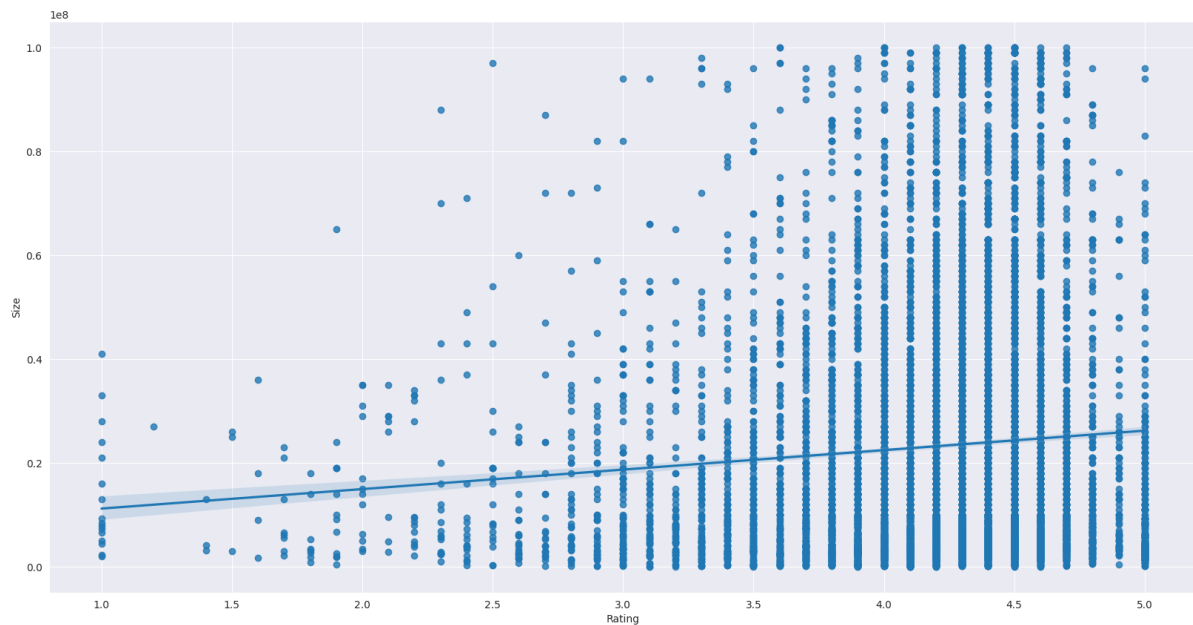


Figure 15 Rating vs size

We can see that apps with higher ratings have more possible sizes compared to apps with lower ratings (<3.0), where the size is almost always under 40 MB.

Is there any relationship between No. Installs and Reviews?

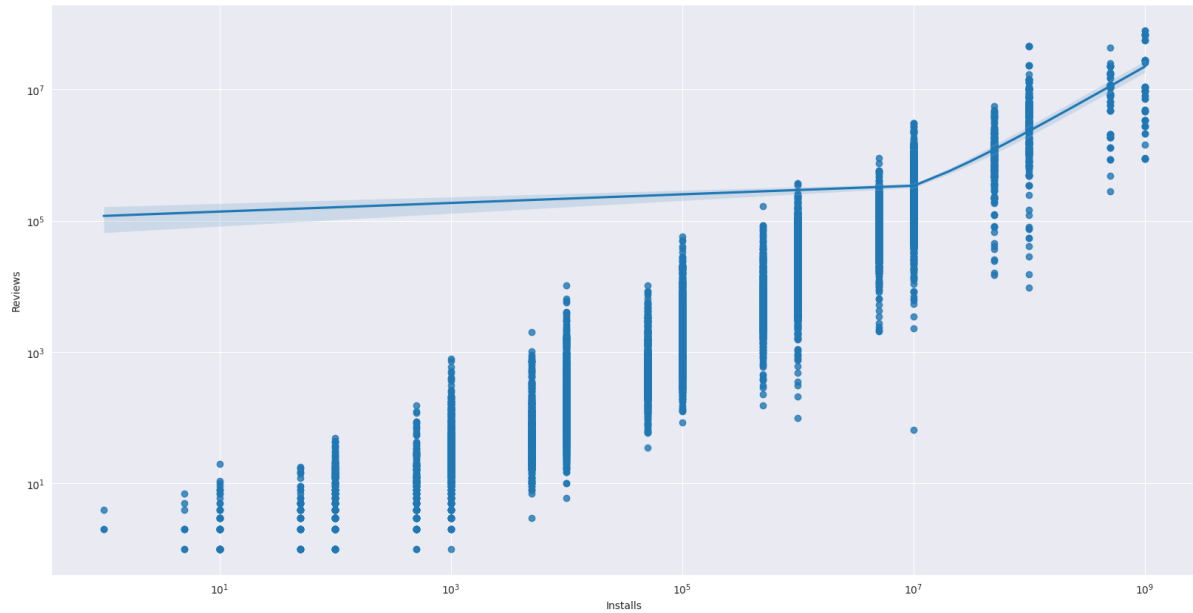


Figure 16 Reviews vs installs

From above plot we can see that apps with more installs tends to have more reviews.

Chapter 5 Models

5. Models

A prediction of Google applications ratings can be done by using the data (for example, size, installs and number of reviews) which were provided by the dataset. Different regression models were used for the prediction such as linear regression, random forest regressor, support vector regression and neural network. Firstly, the dataset was split into two parts, where the testing part (Y variable) will be the rating of the application and it is a continuous type of data. Training data set can be different every time as per the requirements of the model. Every time the data was split into training and testing sets where, training data = 70% and testing data = 30%.

Scikit-learn(sklearn) is an open source and free software machine learning library for python programming language. This library also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities. For this project, sklearn library was used to perform the support vector regressor (SVM), Linear regression, random forest regression and deep neural network (MLPRegressor).

Mean squared error (MSE) was used to measure the squared average distance between the real data and the predicted data. Mean absolute error (MAE) used to measure the absolute average distance between the real data and the predicted data, but it did not pay attention to large errors within the prediction. Overall, MSE and MAE both are used to get regression loss.

R2-score is a score function for regression which calculates the coefficient of determination. The best score for prediction is 1.0, at the same time score can be negative because sometimes the model can be arbitrarily worse.

5.1 Linear Regression

This algorithm used to predict a correlation between one or more dependent and independent variables, and it took numeric values as an input and output. Performance (error rates) of the algorithm depends on various factors including how clean and consistent the data is.

Table 2 Linear regression model result

	Excluding Genres		Including Genres	
	Integer Encoding	One-hot Encoding	Integer Encoding	One-hot Encoding
Mean Squared Error	0.252	0.268	0.264	0.274
Mean Absolute Error	0.251	0.257	0.261	0.263
R2_score	0.013	0.031	0.007	0.030

From the above table, it was very difficult to predict which model (integer/one-hot encoding) gave less error. As we can see, mean squared error for one-hot encoding with genres column gives highest error and integer encoding without genres column gives lowest error. One-hot encoding (excluding genres) is the best option while considering the r2-score. In conclusion, excluding genres column gave better output as compared to including genres column models.

5.2 Random Forest Regressor

Random forest regressor is the ensemble learning method for regression and it will generate multiple decision trees or various subsample of the dataset while training the data and use average or mean to improve the predictive accuracy or to control the overfitting problem. Moreover, decision trees are controlled with some parameters which we can define while creating model code.

Table 3 Random forest regressor model result

	Excluding Genres		Including Genres	
	Integer Encoding	One-hot Encoding	Integer Encoding	One-hot Encoding
Mean Squared Error	0.224	0.227	0.223	0.221
Mean Absolute Error	0.206	0.201	0.205	0.200
R2_score	0.135	0.121	0.138	0.146

Given table describes that one-hot encoding model with genres column is the best choice for random forest regressor model as it has high r2-score and low mean squared error. In conclusion, excluding genres column was not suitable for random forest regressor as it gave high error rate.

5.3 Support Vector Regression

SVR is a version of SVM for regression and it works with continuous values. SVR depends only on a subset of the training data, because the cost function for building the model ignores any training data close to the model prediction.

Table 4 Support vector regression (SVM) model result

	Excluding Genres		Including Genres	
	Integer Encoding	One-hot Encoding	Integer Encoding	One-hot Encoding
Mean Squared Error	0.270	0.235	0.241	0.261
Mean Absolute Error	0.231	0.223	0.231	0.225
R2_score	0.006	0.019	0.018	0.002

From the above table, one-hot encoding without the use of genres column gave the best result as the lowest mean squared error and high r2-score as compared to other models. However, one-hot encoding with genres column gave only 0.002 score and almost 0.261 error rate. After analyzing, dropping the genres column for this model was the quite better option.

5.4 Neural Network

Multi-layer Perceptron regressor used as a deep neural network (DNN) for the rating prediction problem where scikit learn library used to import the model into the python code. The model trains using backpropagation without using activation function in the output layer. Moreover, square error is used as the loss function and the output is a set of continuous values.

5.5.1 Hyper-Parameter Tuning

Result of the deep neural network majorly depends on the different parameters such as number of iterations, activation functions, size of hidden layers and so on.

Number of iterations:

When defined only 2 hidden layers with (100,80) neurons gave very poor results and while defined 3 hidden layers with (100,80,50) neurons gave slightly better results for the model. For this dataset, selection of 4 hidden layers with (100,100,70,40) neurons gave the lowest error rate

as compared to other combinations. Finally, continue with 4 hidden layers as a parameter of the model.

Activation function:

There are several activation functions available, but the most relevant functions are tanh and logistic for this problem. Here, comparison between models done with the help of mean squared error. Below table shows some combinations of the activation function and select the best appropriate option for the model. Logistic function gave high mean squared error as compared to the tanh function. Hence, tanh will be the perfect choice for the model because it gives a low error rate.

Table 5 Comparison using different activation function with models

Models	Activation Functions	
	logistic	tanh
Integer encoding	0.2833	0.2650
One-hot encoding	0.2738	0.2591
Integer encoding (with genres)	0.2813	0.2652
One-hot encoding (with genres)	0.2733	0.2520

Number of Iterations:

This parameter also plays a very important role in the model. Below graph is the analysis of maximum iterations with the average of all model's error rate. The graph describes that when our iteration number was between 25 to 45 it will give the lowest mean squared error and iteration number below 25 gave the convergence warning (or high mean squared error rate). By selecting iteration value 30, model gave the lowest error rate as compared to the other options.

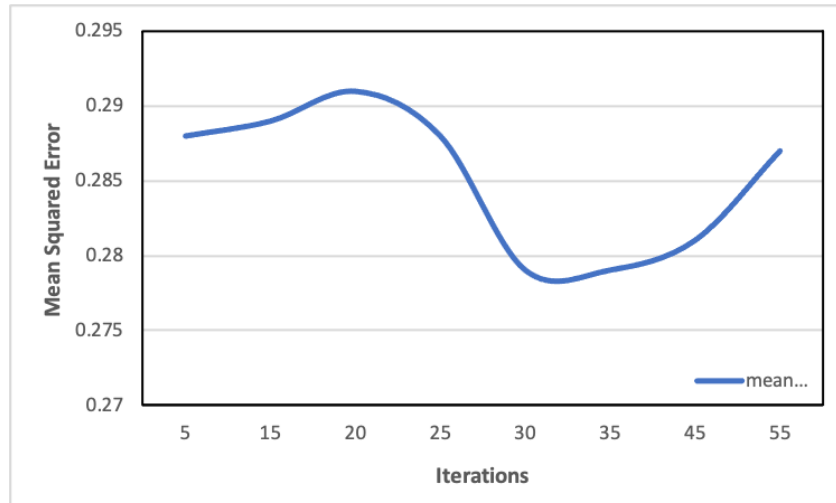


Figure 17 analyzing iterations for model

Final Parameters for MLPRegressor:

Hidden Layers:	(100,100,70,40)
Activation Function:	tanh
Number of Iterations:	30

5.5.2 Result

Table 6 MLPRegressor model (deep neural network) result

	Excluding Genres		Including Genres	
	Integer Encoding	One-hot Encoding	Integer Encoding	One-hot Encoding
Mean Squared Error	0.265	0.259	0.265	0.252
Mean Absolute Error	0.241	0.228	0.232	0.227
R2_score	- 0.017	-0.002	-0.038	0.015

After analyzing the results, the use of a neural network (MLPRegressor) for this problem was the least ideal idea. R2-score for this model gave many negative values, hence it was not a good score for the prediction. However, in terms of mean squared error rate, the one-hot encoding with genres column gave the lowest mean squared error and highest r2-score.

Chapter 6

Comparison of models

6. Comparison of models

Comparison between different models can be done with the help of regression loss and r2-score function because all the models used are the regression models. After performing all possible combinations of models, we can simply differentiate the result with the help of a graph. The model which has lowest mean squared error and highest r2-score number gives the best prediction results.



Figure 18 Comparison of different models

Figure 18, shows that random forest regressor (one-hot encoding) including genres column giving the lowest mean squared error approximately 0.219. On the other hand, again the random forest regressor (one-hot encoding) including genres column gives the highest r2score approximately 0.149 as compared to other models. Overall, it can be stated that the use of random forest regressor (one-hot encoding) with genres column is the best option to predict the application ratings.

However, neural network gave very poor r2-score so, it is not beneficial to use neural network for

this prediction problem. While considering mean squared error for the prediction, support vector regressor with integer encoding (excluding genres column) and linear regression with one-hot encoding (including genres column) gave the highest error rate, hence it was not good for the prediction.

Overall, the result of the test set data gave the lowest error of prediction while using a random forest regressor, meaning that there was very less data which was predicted incorrectly. Moreover, genres column plays a very important role in the prediction process.

Chapter 7

Conclusion

7. Conclusion

The study on predicting Google Play Store app ratings found that the Random Forest Regressor model worked best, giving the lowest error rate when tested with real data. But what was really interesting was how the way we organized the data and included details about the type of app (like genres) affected the predictions. It's important to note that sometimes the predictions might vary a bit. This is because algorithms can react differently depending on where they're used, like on different devices or platforms. So, even if there are slight differences in the predictions, the main takeaway about the Random Forest Regressor being the top choice for predicting ratings stays the same. This shows us that understanding how to organize data and pick the right model is key to making accurate predictions in places like the Google Play Store.

Chapter 8

References

8. References

1. "Scikit learn," [Online]. Available: https://scikit-learn.org/stable/getting_started.html.
2. "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine. [Accessed December 2020].
3. "SciKit-Neural Network," [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html.
4. S. Shaikh, J. Gala, A. Jain, S. Advani, S. Jaidhara and M. Roja Edinburch, "Analysis and Prediction of COVID-19 using Regression Models and Time Series Forecasting," 2021
5. M. R. Putri, I. G. P. S. Wijaya, F. P. A. Praja, A. Hadi and F. Hamami, "The Comparison Study of Regression Models (Multiple Linear Regression, Ridge, Lasso, Random Forest, and Polynomial Regression) for House Price Prediction in West Nusa Tenggara," 2023
6. M. S. Acharya, A. Armaan and A. S. Antony, "A Comparison of Regression Models for Prediction of Graduate Admissions," 2019
7. <https://www.analyticsvidhya.com/blog/2021/06/data-cleaning-using-pandas/>
8. https://www.w3schools.com/python/numpy/numpy_intro.asp