Build a Rating predictive data mining model based on Google play store data (Regression Analysis)

Contents

1.0 Introduction

Nowadays, almost all routine tasks are done by smartphone, and it also saves a lot of time and effort. Developing an android application that can be more convenient and profitable for the entity is the main goal of application developers. However, the competitive and lucrative application market can be more puzzling for application developers. Therefore, after doing lots of research and analysis, the application developer invested in an application that can be more productive and profitable for the long term. People are giving ratings from 1 to 5 as per their experience with the app, where 1 represents the least liked and 5 represents the most liked application. As per the google website, currently, there are over 1 billion active users from different 190 countries [1].

1.1 Problem Statement:

To develop an application, developers need to identify the market needs and requirements as there is a vast market of distinct kinds of applications. In addition, applications should not be complicated to use as per the client's requirement. In the app store, Google is showing application recommendations as per user's usage. However, it does not with idea of developing an app. Building a predictive data mining model which predicts the app ratings as per some important parameters might be helpful to understand market needs and requirements.

1.2 Proposed Solution:

In this project, a data mining prediction model is built to predict the rating of an android application. In the project, a model is built with the help of some important parameters of the application such as size, installs, price, category, reviews, and ratings. Dataset with all these attributes was downloaded from the famous Kaggle website [2]. Data pre-processing steps such as cleaning and transformation were performed on the data set. After preprocessing, different data mining techniques were applied and compared on the dataset. With the help of the Python programming language in Google Colaboratory, data mining techniques are performed and compared [3].

1.3 Description of Dataset

Google play store dataset contains 10841 instances with 13 attributes. Dataset had some missing values and outliers. Description of the dataset as per table 1 [2].

Table 1: Description of Dataset

	Attributes (rows)	Description of Attributes
1	App	Application Name
2	Category	Application Category
3	Rating	Application Ratings
4	Reviews	Reviews given by Number of Users
5	Size	Application Size
6	Installs	Number of installations
7	Type	Application Type (Free/Paid)
8	Price	Application Price for download
9	Content rating	Teens/Adults/Everyone
10	Genres	Sub-category of Application
11	Last Updated	Last updated application date
12	Current Ver	Newer version of app
13	Android Ver	Min. Android version to download the app

2.0 Preprocessing of Dataset

Data cleaning and transformation is the most important part of data mining. Failure to clean the data results in a wrong data mining model.

For detecting Outliers, Pandas library functions: boxplot and histogram are used to detect an outlier [4]. As per figure 1, there was one outlier detected in the rating attribute which was removed later.

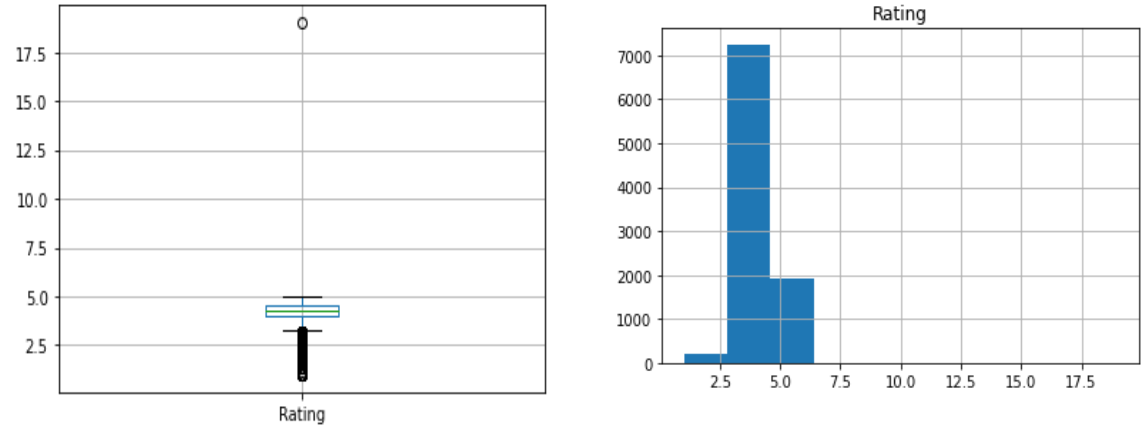

Figure 1: Boxplot and Histogram for detecting outliers

In the data cleaning step, as per figure 2, attributes 'Ratings', 'Current Ver', and 'Android Ver' had missing values. These missing values need to be cleared to clean the data. These missing values are replaced by the means of the total values of age attributes by using the 'SimpleImputer' function [5].

```
App                0
Category           0
Rating          1474
Reviews            0
Size               0
Installs           0
Type               1
Price              0
Content Rating     1
Genres             0
Last Updated       0
Current Ver        8
Android Ver        3
dtype: int64
```

```
App                0
Category           0
Rating             0
Reviews            0
Size               0
Installs           0
Type               0
Price              0
Content Rating     0
Genres             0
Last Updated       0
Current Ver        0
Android Ver        0
dtype: int64
```

Figure 2: Data cleaning Process

Attribute Selection: In our model, we are making a model on selected attributes for greater accuracy of the model. Therefore, a total of six attributes: 'Size', 'Installs', 'Price', 'Category', 'Reviews' and 'Ratings' were selected to build a regression model.

Data transformation: In Regression analysis, data should be in numeric form for accurate results and lower error. In the downloaded dataset, only, 'Rating' attribute is in numeric form. Therefore, by using the lambda function, attributes 'Size', 'Price', 'Installs', and 'Review' were converted

into numeric form. The ‘Type’ attribute was converted into binary data. For the ‘Category’ attribute, two approaches were applied: Label encoding and Dummy encoding (One-hot) [6].

```
playstore.describe()
```

	Rating	Reviews	Size	Installs	Type	Price	Category_2
count	9135.000000	9.135000e+03	9135.000000	9.135000e+03	9135.000000	9135.000000	9135.000000
mean	4.176690	2.493186e+05	21.534957	7.122458e+06	0.078927	1.185283	17.610728
std	0.500816	1.717130e+06	22.593662	4.621827e+07	0.269640	17.365220	7.342431
min	1.000000	0.000000e+00	0.008500	0.000000e+00	0.000000	0.000000	0.000000
25%	4.100000	2.200000e+01	4.900000	1.000000e+03	0.000000	0.000000	14.000000
50%	4.200000	7.420000e+02	13.000000	1.000000e+05	0.000000	0.000000	18.000000
75%	4.500000	2.516900e+04	30.000000	1.000000e+06	0.000000	0.000000	23.000000
max	5.000000	4.489389e+07	100.000000	1.000000e+09	1.000000	400.000000	32.000000

Figure 3: Dataset descriptions after Data Transformation

After performing all preprocessing steps, data was cleaned and ready for the examination of different data mining methods.

3.0 Visualization of Clean Data set

In this section, various attributes and their correlation with each other are discussed. Figures 4, 5, and 6 represent the relationship between Categories attributes and the other three attributes (Number of installs, Price, and reviews). Categories: ‘Communication’, ‘Games’ and ‘News & magazine’ have a larger number of installs. Pricing of application for categories: ‘Family’, ‘Finance’, ‘Lifestyle’ and ‘Medical’ is higher than other categories. The ‘Game’ category has the highest number of reviews [7].

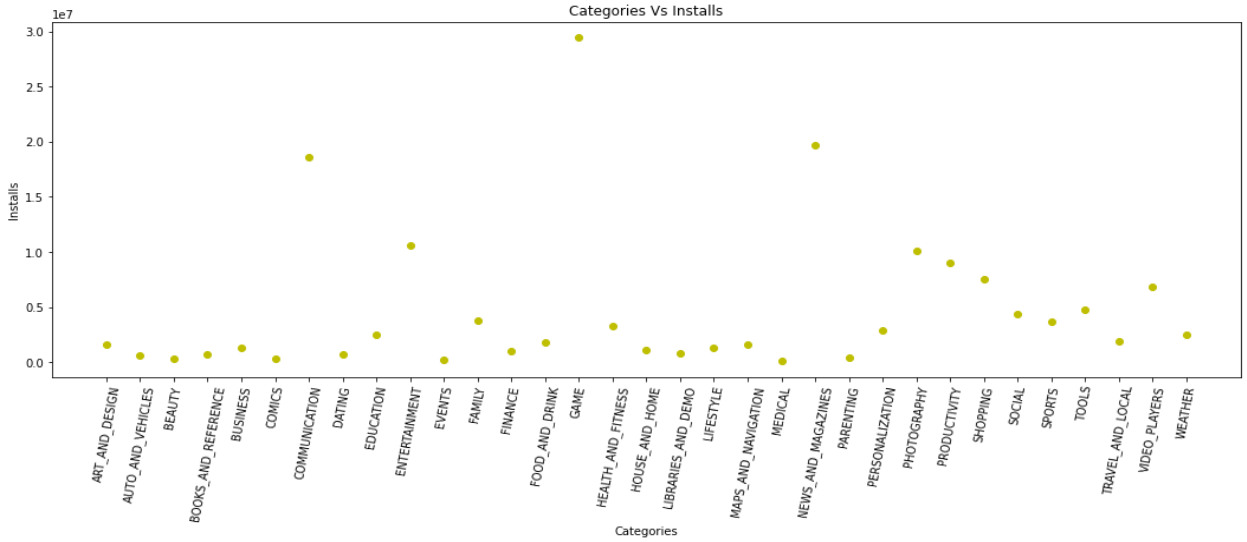

Figure 4: Categories vs Installs

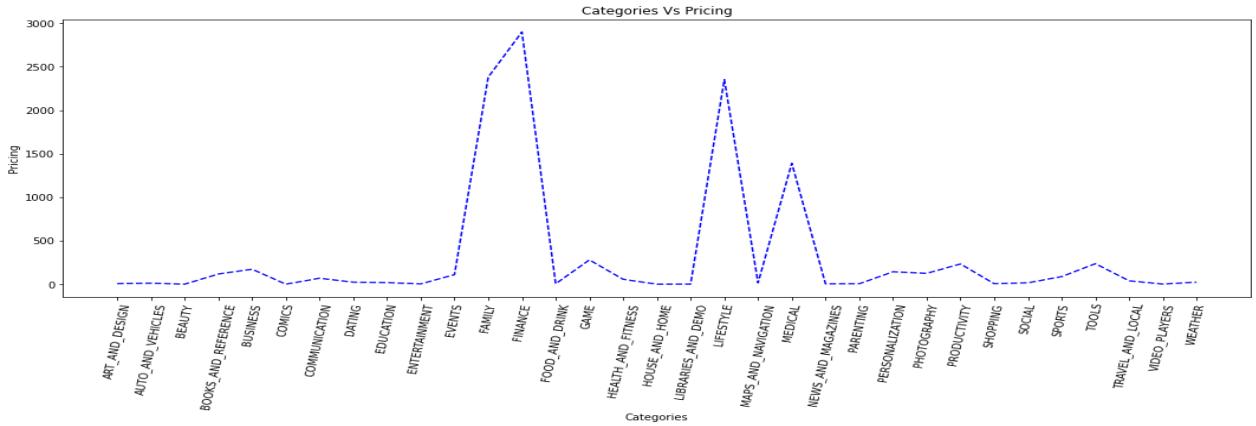

Figure 5: Categories vs Pricing

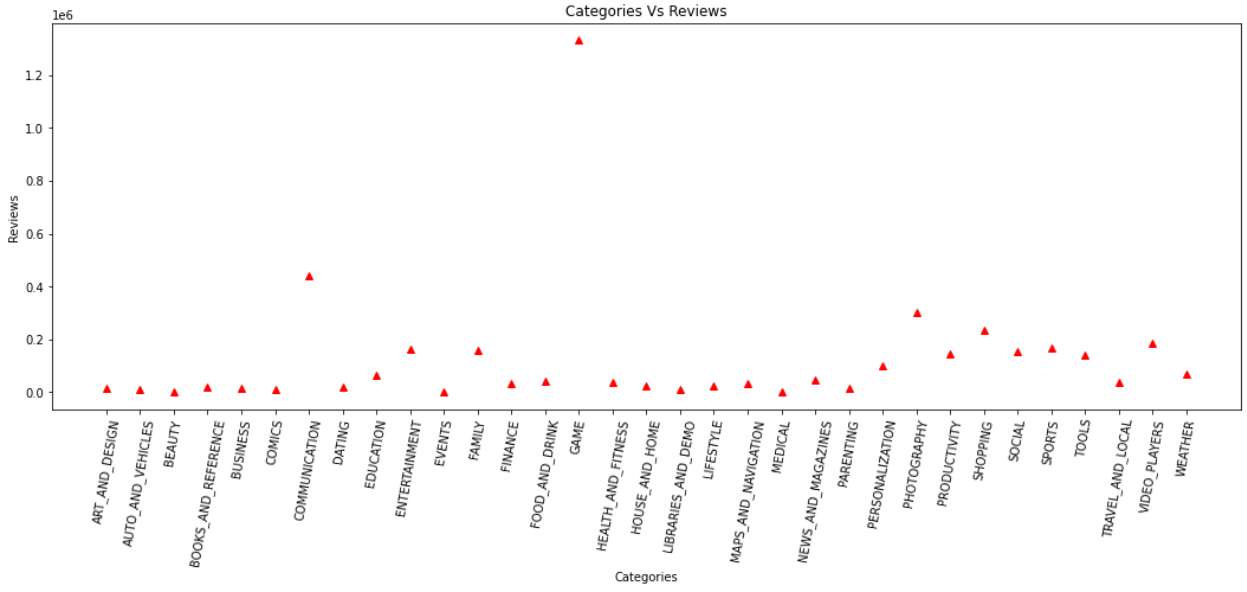

Figure 6: Categories vs Reviews

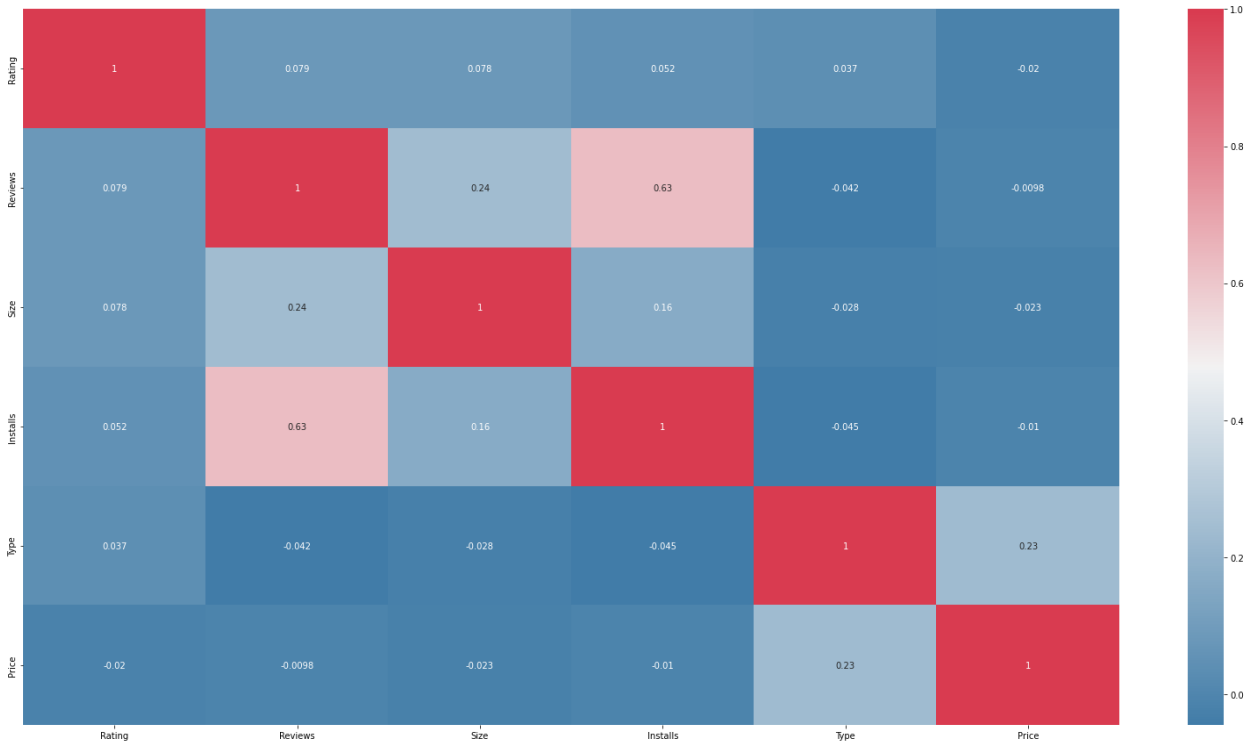

Figure 7: Correlation Matrix

As per fig. 7, attributes review and installs have positive correlations of 0.63. That means the more installation of the application, there will be more reviews [8].

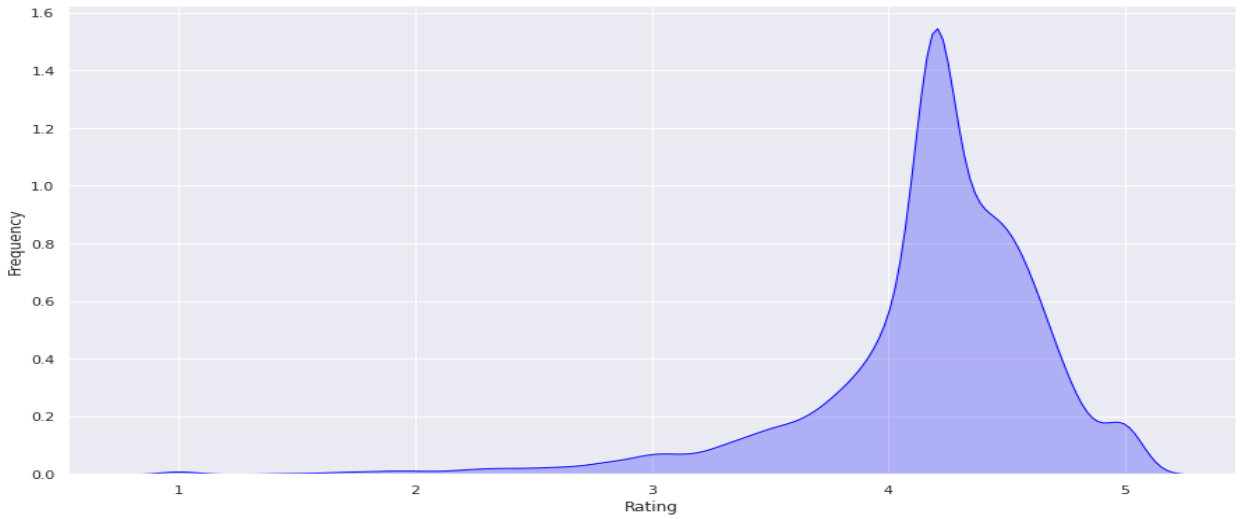

Figure 8: Ratings vs Frequency

As per figure 8, the majority of the google applications have ratings between 3 to 5 [8].

4.0 Data Mining Methods

Different methods were examined and compared on the given data set. Different data mining models such as linear regression, support vector machine, Random Forest regression, and neural network models were applied on the play store data set in this project. Before testing, the given dataset was divided into two separate parts. The 1st set called the training set has 75% of the total dataset and another set called the testing set has 25% of the total data.

As the goal of the project was regression analysis, there were three parameters: Mean squared error (MSE), Mean absolute error (MAE), and R-squared score are considered to measure the accuracy of the model. MSE calculates the average squared distance between predicted dataset values and original dataset values. The lower the MSE, the greater accuracy of the regression technique.

MAE calculates the average of the absolute distance between predicted dataset values and original dataset values. For the higher accuracy of the regression model, the lower value of the MAE is desired [9].

R^2 or Coefficient of determination also known as score function of regression. The desires value of R^2 is 1.0. In the regression model, a higher value of R^2 is desired for better accuracy of the model [10].

These three parameters were tested and compared for better accuracy of the regression model.

4.1 Linear Regression

The 'LinearRegressor' algorithm from the Scikit-learn library is used to perform the linear regression method. In this method, the Linear model is fitted by weight coefficients (w1, w2, w3…. wn) to minimize the distance between predicted and original values in the dataset [5].

Table 2: Linear Regression Result

	Label Encoding Result **(Standardized Data)**	Dummy Encoding Result
MSE (Mean Squared Error)	0.2368	0.2609
MAE (Mean Absolute Error)	0.3285	0.3365
R^2score (Determination Coefficient)	0.0049	0.0168

Data with integer encoding was standardized before testing on a model. As per Table 2, the linear regression model with Label encoding has lower MSE and MAE. However, the model with Dummies has a little bit higher R^2 score.

4.2 Random Forest Regression

Random forest is an ensemble learning technique. In the project, the 'RandomForestRegressor' algorithm from the Scikit-learn library is used for this method. This technique ensemble the various decision trees and enhances the accuracy of predictive models, Moreover, it also manages the over-fitting of data [5].

Table 3: Random Forest Result

	Label Encoding Result **(Standardized Data)**	Dummy Encoding Result
MSE (Mean Squared Error)	0.2159	0.2469
MAE (Mean Absolute Error)	0.3126	0.3215
R^2 score (Determination Coefficient)	0.0927	0.0698

As per table 3, we can conclude that model with label encoding gives lower errors and a higher R^2 score which is desired for higher accuracy of the model. The standardized training data (label encoding data) was used by RFR to evaluate the result.

4.3 Support Vector Regression (svm.SVR)

The 'svm.SVR' algorithm from the Scikit-learn library is used to perform support vector regression.

The dataset with the label was standardized before performing the testing. Table 4 represents the result of this model. MSE and MAE for label encoding dataset are lower and the R^2 score is higher [5].

Table 4: SVR Result

	Label Encoding Result **(Standardized Data)**	Dummy Encoding Result
MSE (Mean Squared Error)	0.2330	0.2645
MAE (Mean Absolute Error)	0.3111	0.3179
R^2 Score (Determination Coefficient)	0.0209	0.0034

4.4 K-Nearest Neighbors (k-NN)

In this method, the value of the new instance is predicted by the closest nearest neighbor of the training set. In the k-NN model, the value of k was set to 15. After applying the k-NN algorithm on the dataset, the results are as per table 5. The dataset with integer encoding has a lower MSE and higher R^2 score compared to the dataset that has dummies [5].

Table 5: k-NN Result

	Label Encoding Result	Dummy Encoding Result
MSE (Mean Squared Error)	0.2231	0.2537
MAE (Mean Absolute Error)	0.3281	0.3340
R^2 Score (Determination Coefficient)	0.0624	0.0443

4.5 DNN (MLP regressor)

Deep neural network algorithm 'multi-layer perceptron regressor' from Scikit-learn library is used to perform testing on the dataset. In this neural network method, many parameters needed to be trained before applying it to the dataset. Hidden layers sizes were set to (100,80) as it gave the lowest error. The value of iteration was set to 50 and the 'tanh' function was chosen as the activation function. All parameters were chosen after examination and comparison with other parameters [11].

Table 6: DNN Result

	Label Encoding Result	Dummy Encoding Result
MSE (Mean Squared Error)	0.2352	0.2645
MAE (Mean Absolute Error)	0.3330	0.3332
R^2 Score (Determination Coefficient)	0.0116	0.0034

The result of this model is described in table 6. This method gives lower MSE and a higher R^2 score with the label encoding dataset. However, this method fails to improve the accuracy compared to other regression models.

4.6 Comparison

By comparing all the above regression model results, we can conclude that the 'random forest regressor' algorithm gives a lower mean squared error and the highest R^2 score. For regression problems, models with the highest R^2 score and the lowest mean squared error give the highest accuracy. The neural network model fails to improve the accuracy. Moreover, the SVR model gives the lowest accuracy for the dummy encoding dataset. Moreover, one more simple Keras

neural network model was tested on the dataset. As it gave almost similar results as MLP regressor DNN, it was not added in the comparison.

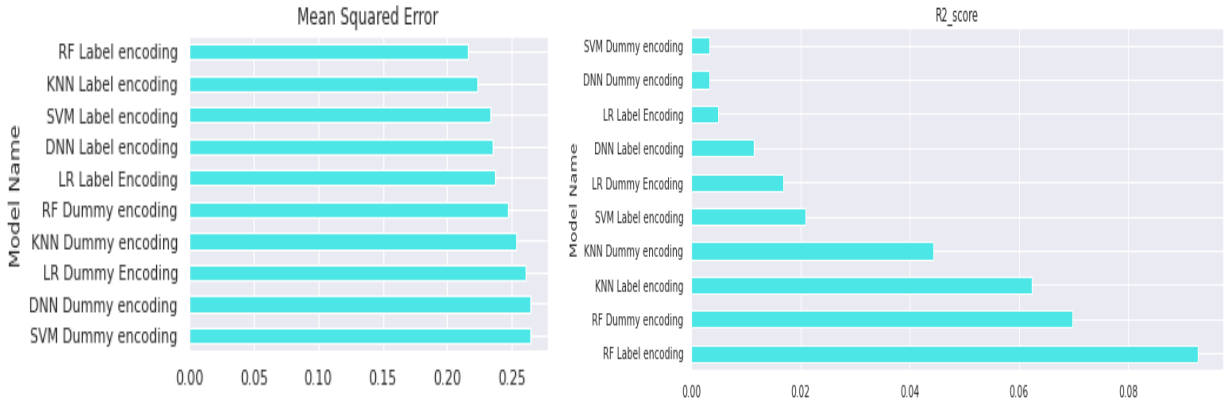

Figure 9: Comparison of different models (MSE & R^2 Score)

4.7 Keras Neural Network

In Python, Keras is a simple and easy library to build a neural network model. Standardized training data was used to perform this model. As values in the dataset were one-dimensional, Dense layers were used to build a neural network. Total two dense hidden layers were used with 128 and 64 nodes, respectively. For these hidden layers, the 'relu' function was used as an activation function. There was also a final output dense layer with one node which used the 'linear' as an activation function. Mean squared error was used as loss function and mean absolute error as metrics function [6] [12] [14]. The summary of the keras neural network model with mentioned parameters is represented in figure 10.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               896

 dense_1 (Dense)             (None, 64)                8256

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 9,217
Trainable params: 9,217
Non-trainable params: 0
```

Figure 10: Summary of Keras Neural Network model

Table 7 represents the MSE and MAE of neural network models. Results are almost the same as the MLP regressor DNN model.

Table 7: Keras Neural Network Result

	Label Encoding Result **(Standardized Data)**
MSE (Mean Squared Error)	0.2328
MAE (Mean Absolute Error)	0.3300

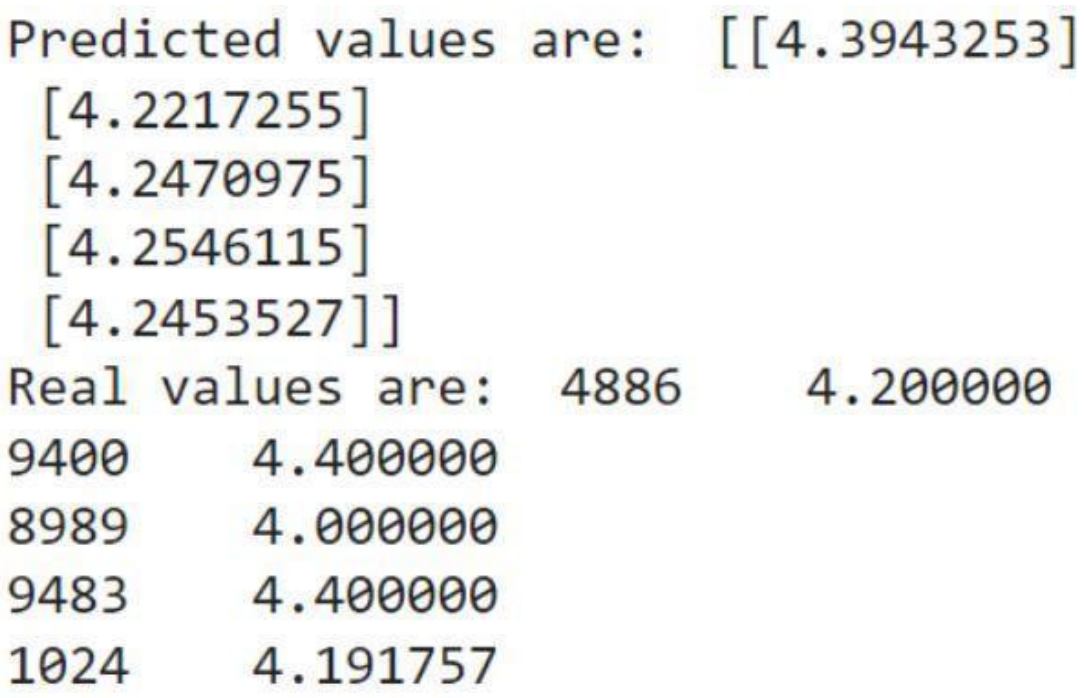

Figure 11: Predicted values VS Actual values (Keras NN)

As per figure 11, we can see that predicted values are very close to real values. Moreover, MSE and MAE are also lower. So, we can say that the accuracy of this model is greater.

Conclusion

In the project, the Random Forest regressor (RFR) algorithm gives the highest accuracy with the dataset with label encoding. As there were many parameters and environments were considered during the evaluation of models so, R^2 score, MSE, and MAE were minorly kept changing during the testing of different models each time. However, the final comparison of the models remained the same. In addition, in the case of neural networks and KNN Methods, it remained unchanged. Two neural network models also give great results but fail to improve the accuracy compared to the 'random forest regressor (RFR)' algorithm. Therefore, the RFR algorithm is the best choice for the regression analysis of integer encoding google play store dataset.

References

[1] “Google Play,” *Android Developers*. https://developer.android.com/distribute (accessed Nov. 23, 2021).

[2] “Google Play Store Apps.” https://kaggle.com/lava18/google-play-store-apps (accessed Nov. 23, 2021).

[3] “Google Colaboratory.” https://colab.research.google.com/?utm_source=scs-index (accessed Nov. 23, 2021).

[4] “pandas - Python Data Analysis Library.” https://pandas.pydata.org/ (accessed Nov. 23, 2021).

[5] “1. Supervised learning,” *scikit-learn*. https://scikit-learn/stable/supervised_learning.html (accessed Nov. 23, 2021).

[6] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*, 2nd ed. O’Reilly Media, Inc, 2019. Accessed: Nov. 23, 2021. [Online]. Available: https://learning.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

[7] “Matplotlib — Visualization with Python.” https://matplotlib.org/ (accessed Nov. 23, 2021).

[8] “seaborn.heatmap — seaborn 0.11.2 documentation.” https://seaborn.pydata.org/generated/seaborn.heatmap.html (accessed Nov. 23, 2021).

[9] A. Chugh, “MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?,” *Analytics Vidhya*, Dec. 08, 2020. https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e (accessed Nov. 23, 2021).

[10] “sklearn.metrics.r2_score,” *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.metrics.r2_score.html (accessed Nov. 23, 2021).

[11] “sklearn.neural_network.MLPRegressor,” *scikit-learn*. https://scikit-learn/stable/modules/generated/sklearn.neural_network.MLPRegressor.html (accessed Nov. 23, 2021).

[12] “Keras: the Python deep learning API.” https://keras.io/ (accessed Nov. 23, 2021).

[13] J. Brownlee, “Your First Deep Learning Project in Python with Keras Step-By-Step,” *Machine Learning Mastery*, Jul. 23, 2019. https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/ (accessed Nov. 23, 2021).

[14] DigitalSreeni, *141 - Regression using Neural Networks and comparison to other models*, (Jul. 14, 2020). Accessed: Nov. 23, 2021. [Online Video]. Available: https://www.youtube.com/watch?v=2yhLEx2FKoY