Android App Rating Prediction Using Multi Layer Neural Network

Noorun Nashfin (email: 160204030@aust.edu)

Kazi Tunaz Zina (email: 160204052@aust.edu)

1) **Introduction:**

   a) **Definition:** With the development of mobile applications (apps) and the expansion of mobile apps markets (e.g., Google Play Store, Apple App Store), predicting user choices and preferences is becoming a difficult work. Different traditional methods are available to predict ratings of android apps. Among them, we choose to conduct our project based on deep learning particularly Multi-layer Perceptron classifier. In our project, we chose to predict the ratings of android apps based on their features. Here, features include app name, reviews given by the users, app size, the number of times apps were installed, app types (whether they are free or paid), content rating(which age groups are installing apps), genres of apps, the last time apps were updated, the current version of the apps and supported android versions. The system can provide ratings depending on various features of a particular app where the traditional rating system is not available.

   b) **Motivation:** Since play store app data has a huge potential to lead app developing business to success, the system can be a helping hand to the developers to choose their market, to choose their target users, to choose app types, and to find sectors for further development.

2) **Related Works:**

   a) In this part, we provide a summary of some works related to our project.

      i) Paper [1] predicts the rating of an app from the features of the app. Their analysis is based on Samsung Android and Blackberry World app stores. In particular, they use Natural Learning Processing (NLP) to extract apps' features from textual descriptions, and apps' features and ratings are used as input.

      ii) Paper [2] assumes that different users like different features of an app. Based on this assumption, they propose a feature-oriented method to transform original ratings into feature data and predict unknown ratings of an app.The predicted user ratings on features can be used to generate the ratings on apps.

      iii) Paper[3] proposes a Weight-based Matrix Factorization(WMF) model which view each user as a document and each app as a word and calculates the weight of each app. The weights are calculated using the term frequency-inverse document frequency(TF-IDF) algorithm. Later, the weights are introduced to matrix factorization to predict app ratings.

b) Differences with the mentioned works: As most of the papers we came across followed machine learning algorithm to predict ratings of apps, while we used multi layer perceptron algorithm to do the prediction. However, we still managed to find some differences. While [1] focuses on extracting features of an app from end-user reviews, we put our concentration on some extra features such as- the number of times apps were downloaded, app version, supported android version, age rating, etc. Again, [3] focused on the dataset collected from Wandoujia(a leading app store in China similar to Google Play Store), our work is solely focused on a dataset based on Play Store.

3) Project Objectives:

b) Tasks of the system:

- **Preprocessing the dataset:** The dataset we used, needed some preprocessing. Here, preprocessing includes adding missing values, removing Unicode characters and null values, dropping bad and idle rows, etc. We represented the preprocessing steps with the help of a flowchart.

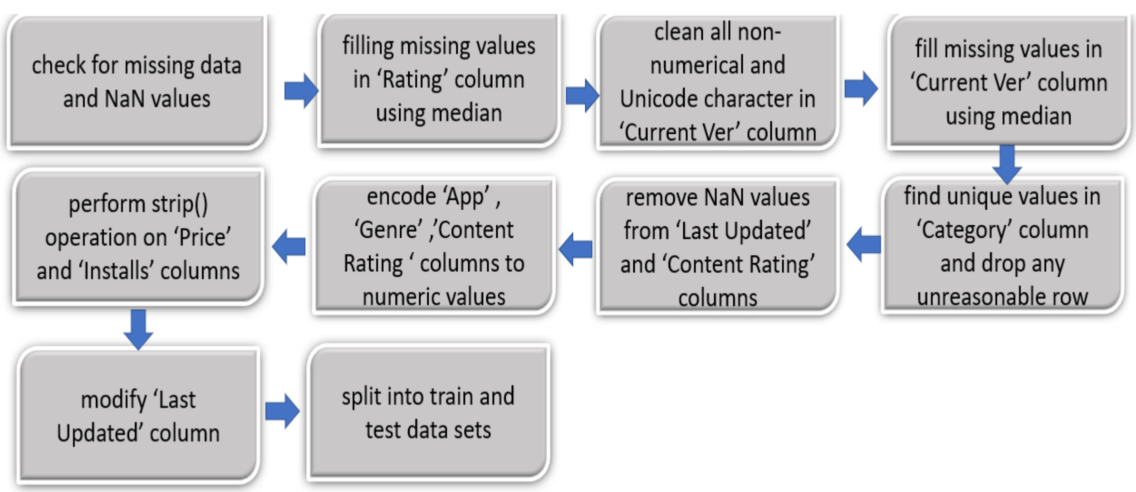

Fig- 3.1: Steps of preprocessing the dataset

- **Dividing into fixed ranges:** We divided the rating attribute into 10 levels , each containing 5 values . The rating attribute column had floating point values so we converted the floating point values into integer values for performing prediction operation. A visual representation of this task is attached below :

```
[ ] for i in range(len(y)):
      if y.iloc[i]>0.0 and y.iloc[i]<=0.5:
        y.iloc[i]=0
      if y.iloc[i]>0.5 and y.iloc[i]<=1.0:
        y.iloc[i]=1
      if y.iloc[i]>1.0 and y.iloc[i]<=1.5:
        y.iloc[i]=2
      if y.iloc[i]>1.5 and y.iloc[i]<=2.0:
        y.iloc[i]=3
      if y.iloc[i]>2.0 and y.iloc[i]<=2.5:
        y.iloc[i]=4
      if y.iloc[i]>2.5 and y.iloc[i]<=3.0:
        y.iloc[i]=5
      if y.iloc[i]>3.0 and y.iloc[i]<=3.5:
        y.iloc[i]=6
      if y.iloc[i]>3.5 and y.iloc[i]<=4.0:
        y.iloc[i]=7
      if y.iloc[i]>4.0 and y.iloc[i]<=4.5:
        y.iloc[i]=8
      if y.iloc[i]>4.5 and y.iloc[i]<=5.0:
        y.iloc[i]=9
    print(y.iloc[0])
```

- **Implementing multi layer neural network :** As our work is mainly based on deep learning, we chose to implement multi layered neural network (MLP).

- **Implementing mean accuracy:** To evaluate our work, we have calculated the accuracy on our testing data set.

- **A comparison with machine learning algorithm :** We tried to compare our deep learning model with a machine learning model by simply comparing their accuracies.

b) Dummy Inputs and Output: In this part, we present some dummy inputs and outputs of our system along with a figure for an easier understanding

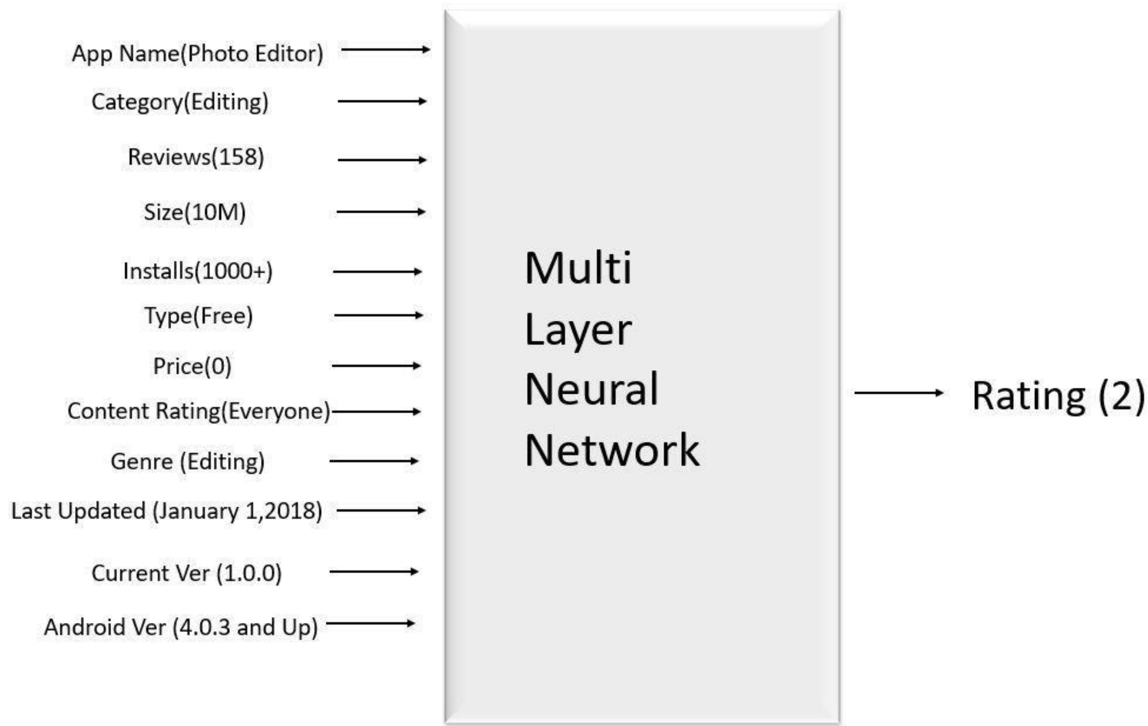

Fig 3.2: Dummy inputs and outputs

The figure above describes dummy inputs and outputs of our system. The rectangular shape represents the classifier. On the left side, there are the inputs. As inputs, we are providing features of an app. The values of the features are mentioned in brackets. On the right side, the predicted rating of an app is the output of our system.

4) **Methodologies/ Models:** As mentioned earlier, we used multi layer perceptron classifier to complete our work. We also used random forest algorithm to compare our work. In this section, we are going to elaborate on the classifier with necessary diagrams.

- **Multi Layer Perceptron Classifier :** Our work is based on deep learning. As MLP classifier relies on an underlying Neural Network to perform the task of classification, we chose to make MLP as our algorithm. It is a feed-forward artificial neural network. An MLP consists of at least three layers of nodes; an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable. We provide the values of the parameters that a MLP algorithm uses , we can definitely tune the value of each of these parameter.

*hidden_layer_sizes=100, activation='relu', *, solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000*

- **Random Forest:** The second classifier used in this work is Random Forest. Random Forest has a decent estimation for regression. Further, it can handle large datasets with high dimensionality. In the following page, we provide a diagram for the Random Forest algorithm.

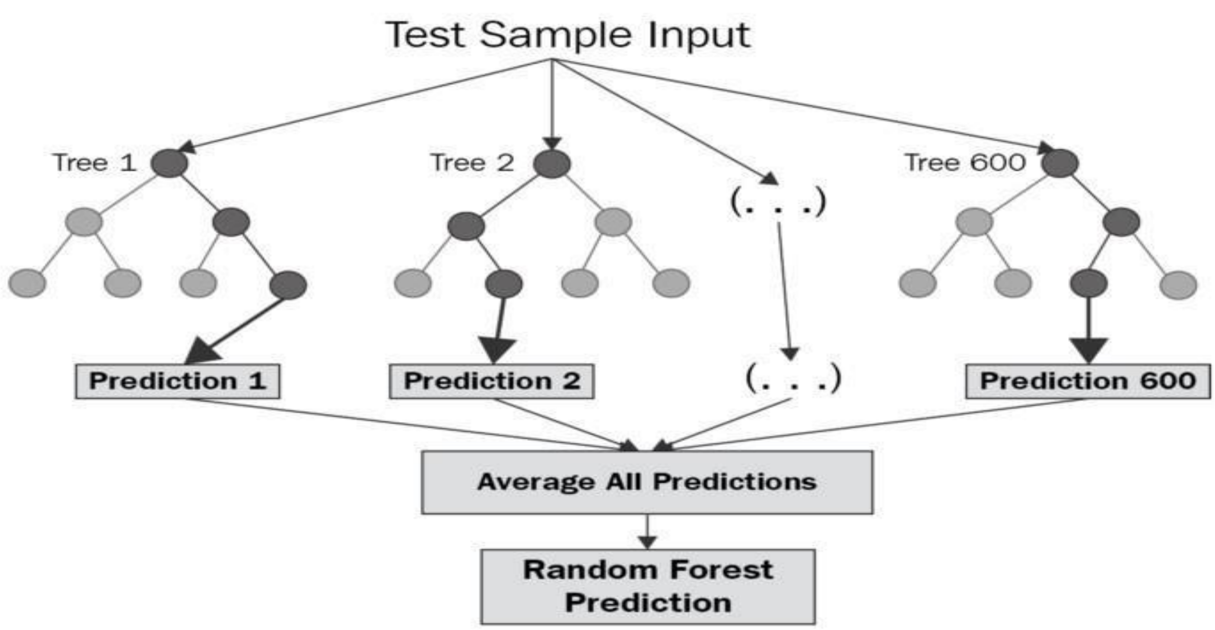

Fig 4.2: Diagram of Random Forest

5) Experiment:

a) Dataset: In this section, we provide some statistical information about our dataset.

i) **Statistic of the dataset:** After preprocessing the dataset we have a total of 39 unique classes. Here, we provide the total number of samples in each class.

CLASS (RATING)	TOTAL NO. OF SAMPLE
0	1695
1	16
2	0
3	0
4	0
5	0
6	488
7	1522
8	5225
9	1894

ii) **Sample from the dataset with classes:** In this section, we provide some samples from the dataset including their classes before we performed preprocessing and after performing preprocessing.

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10836	Sya9a Maroc - FR	FAMILY	4.5	38	53M	5,000+	Free	0	Everyone	Education	July 25, 2017	1.48	4.1 and up
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.0	4	3.6M	100+	Free	0	Everyone	Education	July 6, 2018	1.0	4.1 and up
10838	Parkinson Exercices FR	MEDICAL	NaN	3	9.5M	1,000+	Free	0	Everyone	Medical	January 20, 2017	1.0	2.2 and up
10839	The SCP Foundation DB fr nn5n	BOOKS_AND_REFERENCE	4.5	114	Varies with device	1,000+	Free	0	Mature 17+	Books & Reference	January 19, 2015	Varies with device	Varies with device
10840	iHoroscope - 2018 Daily Horoscope & Astrology	LIFESTYLE	4.5	398307	19M	10,000,000+	Free	0	Everyone	Lifestyle	July 25, 2018	Varies with device	Varies with device

10841 rows × 13 columns

Fig : Before pre-processing

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Current Ver	Android Ver
0	6962	ART_AND_DESIGN	8.0	159	19.0	10000	1	0	1	9	1.515283e+09	1.00	4.0.3 and up
1	2632	ART_AND_DESIGN	7.0	967	14.0	500000	1	0	1	12	1.515974e+09	2.00	4.0.3 and up
2	8656	ART_AND_DESIGN	9.0	87510	8.7	5000000	1	0	1	9	1.533082e+09	1.24	4.0.3 and up
3	7827	ART_AND_DESIGN	8.0	215644	25.0	50000000	1	0	4	9	1.528416e+09	0.00	4.2 and up
4	7022	ART_AND_DESIGN	8.0	967	2.8	100000	1	0	1	11	1.529453e+09	1.10	4.4 and up
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10836	8173	FAMILY	8.0	38	53.0	5000	1	0	1	39	1.500941e+09	1.48	4.1 and up
10837	4609	FAMILY	9.0	4	3.6	100	1	0	1	39	1.530835e+09	1.00	4.1 and up
10838	6891	MEDICAL	8.0	3	9.5	1000	1	0	1	71	1.484870e+09	1.00	2.2 and up
10839	0	0	0.0	0	0.0	0	0	0	0	0	0.000000e+00	0.00	0
10840	9486	LIFESTYLE	8.0	398307	19.0	10000000	1	0	1	67	1.532477e+09	0.00	Varies with device

10840 rows × 46 columns

Fig : After-preprocessing

iii) **The split of the dataset:** We split the dataset according to the following manner.

- **Train and Test:** The train and test dataset were assigned 70% and 20% data of the whole dataset respectively.

- **Validation fraction:** To check the validity of our model, we assigned 10% data for validation fraction.

b) **Evaluation of Model:** We tried to evaluate our proposed system by checking the accuracy score on the test dataset of our model.

c) **Results:** We tried with four different settings . The results along with the **ablation experiment** based on the hyper-parameters is given below :

Model	**Iteration**	**Hidden_layer**	**Accuracy**
01	300	100	40.18%
02	3000	100	65.22%
03	10,000	100	65.22%
04	3000	300	65.31%

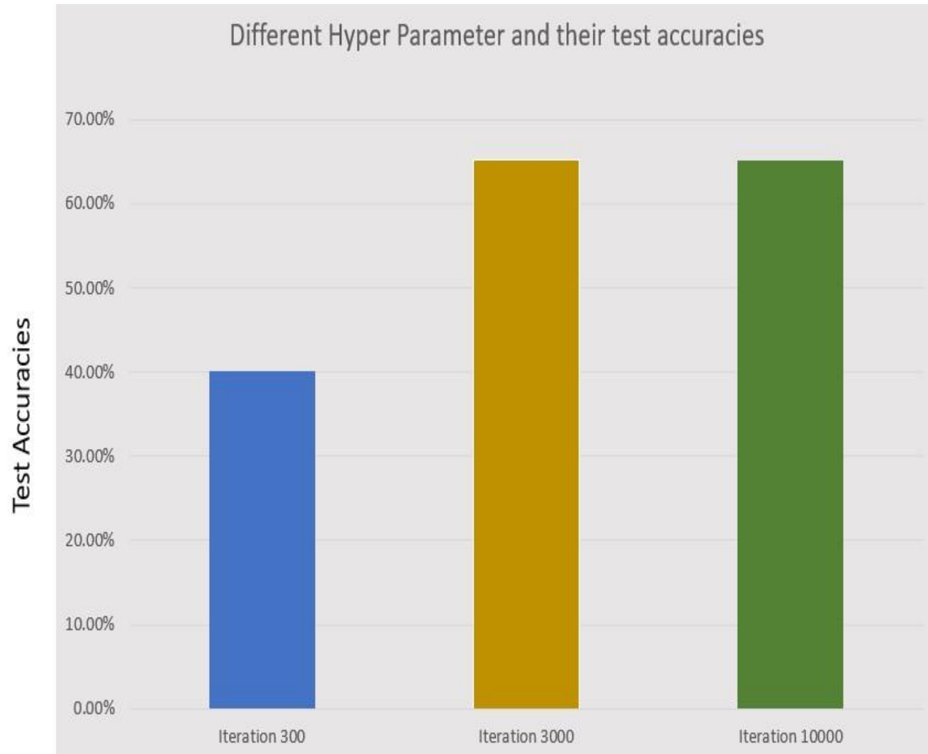

Fig : Different hyper parameters and their test accuracies

A Comparison with machine-learning algorithm : We have performed a machine learning algorithm to check if we can get a better accuracy . We performed random forest classifier and got 93.78% accuracy . Here we tried to show the comparison with a graph :

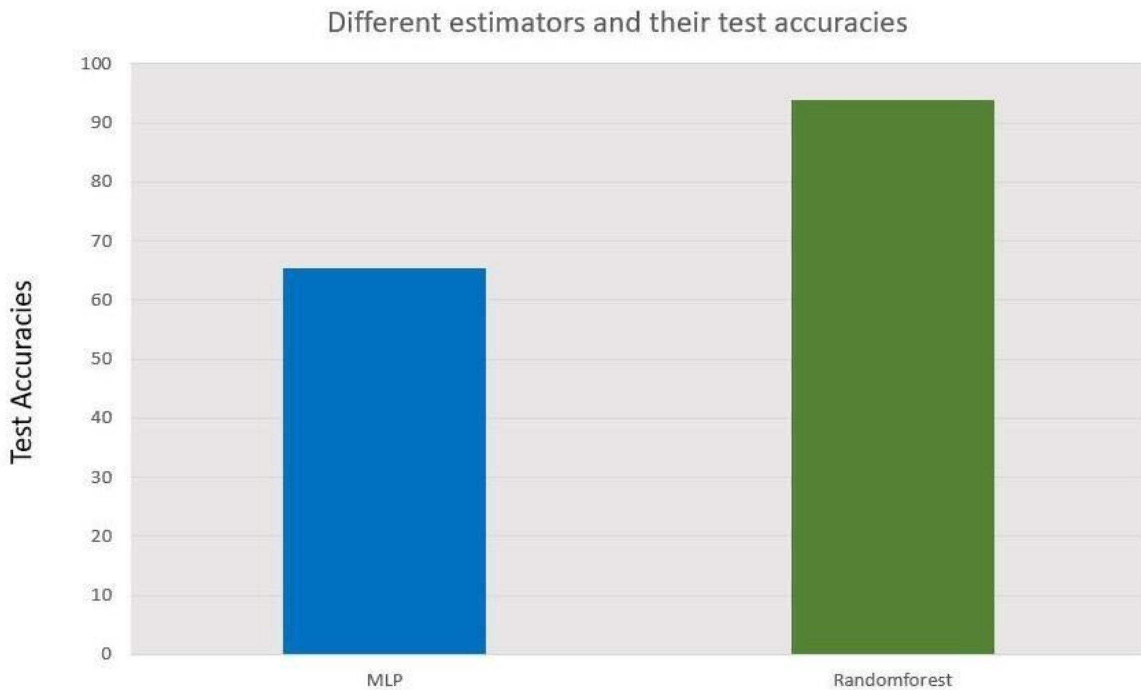

Fig : Different approaches and their test accuracies

Limitations:

- Small dataset : Neural network performs well with larger datasets. We have a dataset of 10,840 rows where only 70% was used for training the model. So the model gave excellent results with machine learning algorithm rather than deep learning algorithm.

- Dividing into fixed ranges : We divided the rating column into 10 groups each containing 5 values. The accuracy would have increased if we took only 2/3 values in each group.

6) **Conclusion:** Since our work is based on deep learning, we tried to implement multi layer perceptron algorithm for regression problems. We have achieved an accuracy of about 65.31%. Further, we implemented machine learning algorithm and achieved an accuracy of about 94% by using Random Forest .We can consider the achieved accuracy a satisfactory one. There awaits more for us to learn and explore.

References:

[1] Sarro, F., Harman, M., Jia, Y. and Zhang, Y., 2018. Customer Rating Reactions Can Be Predicted Purely using App Features. *2018 IEEE 26th International Requirements Engineering Conference (RE)*,.

[2] Liang, T., Chen, L., Ying, X., Yu, P., Wu, J. and Zheng, Z., 2017. Mobile Application Rating Prediction via Feature-Oriented Matrix Factorization. *2017 IEEE International Conference on Web Services (ICWS)*,.

[3] Meng, J., Zheng, Z., Tao, G. and Liu, X., 2016. User-Specific Rating Prediction for Mobile Applications via Weight-Based Matrix Factorization. *2016 IEEE International Conference on Web Services (ICWS)*,.