CMME139

Introduction to Deep Learning

Assignment #1

Due Date: 23/05/2023 Tuesday (23:59)

Score Contribution: 20%, evaluated over 20 points.

The ideal programming language to use in the assignment is R (expected programming language).

The code submission is expected to be in R.

However, it is also possible to use other programming languages (such as Python). However, in such cases, our support will be minimal, and we also ask you to provide instructions for how to run, test and debug on top of the assignment.

Objectives

- **To have a thorough understanding of the dataset to be used in the course.**
- **To perform feature engineering on the dataset**
- **To apply traditional machine learning techniques to the dataset as a baseline**
- **To have experience on the data science workflow**
- **To learn to use some commonly used software libraries for data mining and machine learning**

Structure of the Assignment

This is a group assignment. Students are expected to form groups, communicate the group formation to us via Canvas (by enrolling into one of the pre-created groups on Canvas). The groups will be composed of 3-4 students.

Overview

In this assignment (and in the upcoming assignments), we will be working with a publicly available Kaggle dataset named “17K Mobile Strategy Games” which is a Web-scraped data from Apple Play store. This is the data of 17007 strategy games on the Apple App Store. It was collected on the 3rd of August 2019 (supposedly! But there are contradictory entries to that), using the iTunes API and the App Store sitemap. The dataset is accessible via the following URL:

https://www.kaggle.com/datasets/tristan581/17k-apple-app-store-strategy-games

The Web site contains a brief overview of the dataset, and several statistical analyses of the attributes of the data. The Kaggle Web site also contains several exploratory code analyses, several of which are implemented in R. (Which may be useful for you when exporting and processing the data).

In the course of the work, we will also use the following 2022 publication as a starting point:

https://link.springer.com/content/pdf/10.1007/s00521-022-07154-z.pdf

The paper and the csv format of the dataset is also available on Canvas, under the "assignment related documents" tab. Although the dataset is an extensive Web scraping study, it contains unnormalized/unsanitary data and for some games, several of the fields are missing. Additionally, the dataset contains (possibly) useful information such as high resolution iconography of the games, which can be useful in a machine learning study.

As explained in the course objectives, the purpose of this assignment is to go over a machine learning workflow, clean the data, and develop some baselines for the upcoming assignments using older traditional machine learning techniques. Note that our objective is not replicating the results of the study: I am sure once you read the paper, you will realize that some of the quality metrics are not described clearly, and re-inventing them is out of the scope of this assignment. It is also possible to generate several (possibly useful) new features from the dataset.

Specifications of the Assignment

Here, the tasks in the assignment are provided in a step-by-step manner:

Task1: Importing (0 Points)

Download the data, extract it, and import it to RStudio (or any software that you are using) as a data frame.

Look at the dataset metadata.

Check out the paper to develop an understanding of what is in the dataset, and the problem the paper is examining.

When you examine the dataset, you will realize that there are 18 columns in the dataset. However, most of these columns either contain unusable or compound information. Thus, we need to clean the data and separate useful components of the compound information.

Task 2: Data Cleaning (5 points)

Several of the columns contain unusable information, hence can be removed from the dataset in a data mining (or machine learning) task. Specifically:

URL and ***name*** attributes do not contain mineable characteristics. Remove them*

*Note that the paper also removes other columns. We will have some uses for them later. So, we are keeping them for the moment.

Subtitle*:* Create a new (binary) column. The new column will contain 1 if the game has a recorded subtitle, and 0 otherwise. Remove the original subtitle column after this.

***Icon URL**: Remove* the Icon URL column, but only after reading task 4!

Average User Rating**, **User Rating Count**, and **Price columns are clean and useful columns. Keep them.

In-app Purchases (IAP): After you read the paper and check out the dataset metadata, you will realize the in-app purchase column is problematic: There is no documentation about it, it contains several values, and many have missing values. We will use the prescription of the paper for extracting information in this column: Calculate number of ***IAP values (counts)***, ***minimum IAP, Maximum IAP, Sum IAP***, and ***Average IAP*** and add these aggregates as new columns. For missing values, we assume they contribute nothing. (I am not saying 0, since I do not want to count them in number of IAP values count)

Description: In this assignment, we will not do any text analytics. So, description column has minimal value to us. However, count the number of words in the descriptions and add the counts as a new column. Next, remove the description column.

Developer: For each developer find the number of games they have developed in the dataset. Following the paper's prescription, create 2 categories. (I will not spell out this specification in the assignment, check the paper!)

***Age Rating**:* Keep the already categorical age rating. However also create a second age rating attribute, which categorizes games as "4+" and "9+". (You can also propose different partitioning strategies here. All will be welcome if you describe your reasoning)

***Languages**:* We propose two attributes. First, following the paper, create a binary number of languages attribute having two values: single and many. Second create a binary Is_Available_in_English attribute ("No" and "Yes" or 0 and 1).

Size: Keep it as it is. (Note that it is also possible to create an attribute such as small versus large. It may even be a better decision to create categories of sizes. Feel free to do this or not)

Primary Genre: Remove it. We agree with the paper: It is too skewed to be useful in any way.

Genres: Following the paper, create a ***number of genres*** attribute. Follow the paper's prescription. Note that there is information loss in this representation: We are losing the commonalities between games and their respective genres. (However, creating on-hot variables may require too much work compared to their relevance to the course)

Dates: Create two new attributes: ***Release month*** (which month the game has been released, extractable from original release column), and ***elapsed months*** (how many months have passed since the current version release date, examined at 03.08.2019, data collection date. You should solve any problems this decision creates).

Additional Attributes: Add 2 additional attributes:

- ***Game Free***: Whether game has a value in IAP column or not. 1 if no values, 0 otherwise.

- ***Categorical Rating Count:*** Binary: “Low” and “High” low meaning lower than **median** rating count in the dataset, high otherwise. (This will be used as the objective function in our study. **That is, we will try to predict this**)

Task 3: Missing Values, Formatting (4 points)

Some of the attributes will have missing values. There can be three methods to deal with this problem:

- You can remove the game containing the missing value
- You can impute (fill-in or replace with something) some value to the missing value.
- You can remove the attribute. (That is an alternative of last resort, but if an attribute is overly skewed and shows little connection to the objective function, one can decide to do this)

For example, the paper describes that for languages and average user ratings, the authors impute values. That is a reasonable technique, since otherwise you are at the risk of losing too much of the data (and there is a reasonable explanation to why they resort to filling in those values). However, carefully consider the consequences of each of your imputations.

Read the paper and decide on how you would like to deal with missing values in your data.

Check your data frame for any other missing values. Remove or impute with your own discretion. (Make sure you include what you did as descriptive comments in your code). Before starting to apply machine learning algorithms our data should be cleaned completely of missing values (in one way or another).

Next, check your data frame columns. Do not take any risks: convert anything that is categorical into categorical data (That is defined as factors in R. Any character-based data is by nature categorical, but any numeric data can be also converted into categorical data using the “as.factor” method)

Task 4: Prepare yourself for the Next Assignment (2 points):

Icon URL: The column stores a URL for each game, which stores a 512x512 jpeg image of the game. Since icons are of high quality, we will not work with all of the games in this course. Select the 200 games with the highest user ratings, and 200 games with the lowest user ratings.

Write a code to download the images of your selection and store them on your computer. We will not be using these images in this assignment, but it will save you time and inconvenience in the later assignments. (I suggest using the ID field as the names of the icon jpeg's to be able associate the, with the instances in the csv's. For example, icon of the game with ID 28921427 can be stored as 28921427.jpeg for convenience.) Remove the Icon URL column afterwards.

Note: Even the image data for 400 games is quite large in volume. This is not an essential requirement of the assignment; Therefore, include a short, commented out segment in your code that downloads a small sample of the jpegs (such as first 10) for us to test your code.

Task 5: Dataset Partitioning (1 points)

Before continuing, drop uninformative attributes from your data frame (such as ID). Check out your attributes and make sure you are not representing anything multiple times (This is the easiest way to introduce bias into your data. 2 attributes with the same information means that information is twice as important, especially when giving automated predictive decisions)

I suggest you develop two tests. One with all the above attributes, and one after removing the user rating count attribute (with the user rating count it would be so obvious, isn’t it?). The purpose of this task is to create baselines. Competency will be evaluated not the performance. Pick whichever you would like to continue in the later assignments. (I would select without the user rating count)

Divide your dataset into two parts: A training set and a testing set. 80% of the data will be in the training set and the remaining data will be in the testing set.

Task 6: Scale your attributes (if needed) (3 points)

Scale any of the numeric attributes in a range such as [0 .. 1]. This can be done manually, by finding the range and dividing the values to range value. This can also be done using the **scale** function in R. (Learning opportunity! Google it! The function scales to range [-1..1], but that would work equally well). Our objective in scaling is to remove more bias out of the framework: large values equate to larger similarity distances, hence when attributes are not competing on equal terms, it is very easy for the machine to learn, similarities are dominated by larger-valued attributes. Again, it is a bias issue.

Alternatively, check the code examples (of Task 7), where you will find a use case of the scale function.

For convenience, we provide a small function that can do the task for you:

```
range <- function(x){ (x-min(x))/(max(x)-min(x)) }
```

Task 7: Creating Baselines for Evaluation (5 points)

In this task, you will evaluate the predictive performance of the data and the objective. We will experiment with 3 algorithms that we have covered in Session 1:

- Naïve Bayes
- K-NN
- SVM (Support Vector Machine)

For this purpose, we will use the “e1071” package in R, which supports several machine learning algorithms and provides easy-to-use interfaces for them. Below, you can find documentation for the package:

https://cran.r-project.org/web/packages/e1071/index.html

https://cran.r-project.org/web/packages/e1071/e1071.pdf

We will use Naïve Bayes (page 35), SVM (page 52) and GKNN (page 20) in the package. Note that some of these functions can also scale attributes using their own parameters.

Additionally, we will provide a short R script on Canvas and do hands on coding in Session 2 Workshop, that uses the algorithms for the “iris” dataset, available as a sample dataset for all R installations.

Test your baselines, include your evaluations as comments within your code.

What to Submit

No reports accepted! Only code. Annotate your code so that anyone can understand what you are doing through your code. That is the reason we ask you to include baseline evaluations in your code. No need to send data frames or data itself, we will run your code.

Note: include all “include.packages” instructions in your code. But please comment them out, since otherwise we will need to install the same packages for every group (again and again, and that sometimes raise conflict problems)

Note: include all library inclusions.

You will be asked to submit your code through Canvas.

What will be graded

- Code correctness
- Code Clarity (a.k.a quality)
- Understandability and annotation quality
- Decision Making process and suitability of decisions in Tasks (for intentionally vague segments)
- Sufficiency in tasks
- Timeliness

Some Tips and Tricks

Save your progress in multiple data frames: long programming tasks tend to be messy. Especially if you are working with large data and do a lot of processing. If you want to change something, saving your progress at intermediate steps can be a life saver.

Divide the tasks: Do not forget this is a group project. One idea is to divide the attributes among group members; each group member works on their respective tasks; and the group merges the jobs later. The ID field can be utilized to align multiple people’s tasks.

About working with people: This is a group assignment. We expect you to work with others, not carry others on your back. The only report we will accept is about imbalanced group work. If you (collectively) feel that some group members are not contributing enough, make sure to communicate this with us.

Do examine the data! There are many problems with data that I have not mentioned. Although none of them would influence the results marginally, and that we are not grading for accuracy, you will be surprised how many problems this simple dataset contains.

Final Note

Preprocessing is one of the most grueling parts of the Data Science workflow. Hence, it is possible that Assignment #1 will be much heavier than the later tasks (It is highly likely to get single page assignments once everything is rolling later in the course!). I suggest you start early for this one.

Good Luck!

I hope you will enjoy this assignment

DL TEAM

Appendix – Some Useful Functions and Libraries

- summary( )
- apply( )
- is.infinite( )
- is.na( )
- is.nan( )

Library(reshape2)

- mutate( )
- select( )

Library(dplyr)

- arrange( )
- group_by( )

Library(jpeg)

- download.file( )
- writeJPEG( )

Library(tidyverse)

- str_split( )