University of Pavia

Data Science and Big Data Analysis

Final Report of Big Data Analysis

– Steam Game Reviews –

Author:
Naoya Kumakura
Lucas Ronquillo Bernáldez
Sofía Fernández Coto

document submitted in fullfillment of the requirements for the

Final Assignment

in the

Department of Computer Engineering
Computer Science

1 Introduction

later write this

2 Dataset and Hypothesis

In this section, we report about the dataset that we used and our hypothesis.

2.1 Dataset - steam reviews

We used dataset "steam reviews" on kaggle[1]. The Steam gaming platform is one of the largest and most active online video game stores, where players not only buy games but also leave reviews and feedback after playing. This dataset contains a large number of user-written reviews (text), review ratings, vote counts. It is particularly suitable for projects that involve Natural Language Processing (NLP), as each record includes a free-text review that allows for a wide range of linguistic and sentiment analysis. The size of the dataset is 2.16 GB, 5 columns (Table 1).

Table 1: title

app ID	app name	review text	R_score	R_votes
10	Counter-Strike	ruined my life.	1	1
10180	Call of Duty*	good until hackers ruined it.	-1	0

* Truncated; Call of Duty: Modern Warfare 2

2.2 Analysis of Dataset

Using Numpy and Hadoop, we investigate the features of this dataset.

2.2.1 Analysis with Numpy

with numpy, we noticed these fact;

- **Missing Data** — 53 missing review text. Other infomation are completed
- **Similar Expressions** — (e.g. ":)", "great game!")

Although the current dataset does not contain missing rating data, building a model that can infer sentiment from review text remains important for handling potential future gaps. Reviews containing recurring phrases like ":)" or "great game!" are likely to be less informative. As such, this redundancy can serve as a valuable feature for models aiming to separate standard reviews from novel ones.

2.2.2 Analysis for Review text with BoW

We noticed that the dataset is required for preprocessing, about these features:

- **Number expressions** — Pure numbers are filtered; meaningful ones (e.g., 10/10) should be whitelisted and kept.
- **Typo** — Minor errors should be autocorrected; severe ones would be filtered or mapped to [UNK].
- **Game terms** — Retained via whitelist(e.g. nerf, FPS)
- **Emphasis** — Normalized (e.g., baaaaaad → bad); kept as +emph if

[1] https://www.kaggle.com/datasets/andrewmvd/steam-reviews

sentimentally relevant.

2.3 Hypothesis

Based on these analysis mentioned in previous section, We made 4 hypothesis about this Dataset.

(I) an apple

(II) Logistic Regression Model for Review score completion

(III)

(IV) a durian

(II) Logistic Regression Model

This project involves constructing a Bag-of-Words (BoW) model and performing logistic regression. The process begins with BoW construction on Hadoop, where specific expressions (e.g., 10/10, ****) are predefined using regular expressions or a dictionary. The BoW is then generated via MapReduce and stored in HDFS. Subsequently, data loading and preprocessing are performed using PySpark. This involves loading the BoW word dictionary to create word vectors, followed by feature transformation (TF or TF-IDF). Finally, logistic regression for training and inference is conducted, leveraging PySpark MLlib.