Shilling Attack in Recommender System

Shivam Kumar

shivamkumardss2018@gmail.com

Cluster Innovation Centre, University of Delhi

We all enjoy watching movies, don't we. We watch movies based on their reviews, but nowadays Filmmakers use fake reviews to promote their movies or Someone can use fake reviews to demote a movie. In this research I have worked on a Project for the detection of fake reviews. I used Machine Learning for this purpose. In this research paper, I used three models for better, reliable and accurate prediction. By offering a through analysis of shilling attack models, detect attributes, detection technique, this research seeks to close this gap. This research give precious intuition into shilling attack models, detection attributes such as: tittle,genre,director,actor,rating, of the movies and detection algorithms such as:RF, KNN Model. Our main objective is to identify the fake user and genuine user in movie recommender system on the basis of rating and other features.

Keywords - Recommender System | Context Based Filtering | Random Forest | KNN | Voting Regressor

1. Introduction

1.1 History

These days whether you see or observe a movie, you are getting recommendations for more content to view, such as the rating of a movie. The Recommender System is susceptible to deceptive advertising. Attackers exploit fabricated user-generated content data, such reviews and ratings to rig recommendation rankings. Reducing the likelihood of attacks in recommender system is crucial for preserving sustainability and fairness(1). The information is filtered through context based filtering recommender systems. The purpose of this study is to assess the effectiveness of machine learning algorithms in discerning between Genuine rating and Fake Rating.

1.2 Findings

A movie recommendation system also known as a movie recommender system, uses machine learning to filter or forecast a user's preferred movies based on their browsing history and activity. Context based filtering includes user contextual information in the recommendation process. Context-based filtering in a movie recommender system involves considering additional contextual information beyond just user preferences and movie features. This context can include various factor such as tittle of movie, Genres of movie, time duration of movie, rating of movie or any other relevant information that might impact a user's movie preferences in a specific situation or context(2).

1.3 Highlights

There were set of models which could have been used for this project. Some of the models that have been used in this project are as follows:

1. Random Forest Model
2. KNN Model
3. Voting Classifier

1.4 Impact

This project will help suggest movies to the user as there will be real ratings based on title, cast, director, genres of the movie rather than being based only user rating. Thus improving the models accuracy. This model is also capable of predicting upcoming movies rating based on the title, cast, director and genres.

2. Background and Context

In order to affect rankings, shilling attacker introduce fake user-generated content (UGC) data into recommender systems. Many people rely on these recommendation systems. To affect the ratings, attackers generate fake profiles and insert them into rating matrix. There are three types of detection system- supervised, semi-supervised and unsupervised. Much work has been done in this field. The main focus of these papers was detecting a shilling attack using various methods, most of them generally based on user credibility and rating time series. Various methods were proposed as there are many problems while detecting shilling attack, it is difficult to detect the user profiles that are attacking, and there is also a lack of optimization. To predict the fake users, they used the user's score, average ratings, and their similarities. Using these prediction shifts i.e the change is the rating scores in the predicted value, and after the implementation of an attack where the attack size and filler attacks vary. Although their results were good but most of them only considered or used the datasets consisting of only users and rating of the movies. These datasets lacks features that can be used or train machine learning, so they rely mostly on a statistical approach.

3. Methodology

3.1 Datasets

We have searched and collected the movie rating datasets from various platforms such as-movie lens, Kaggle and Github. But In this research paper we would use movie rating datasets which is collected from Kaggle and Github. We would definitely agree that the movie rating datasets which is collected from Kaggle and Github that datasets is a genuine datasets. In this research paper we would be use the two types of datasets that is first is genuine datasets which is collected from Kaggle and Github and second is fake datasets which is generated by me through Python code. The datasets is based on various features like Tittle of the Movie, Genres of the Movie, Actor of the Movie, Director of the Movie, Duration of the Movie, Rating of the Movie(3).

3.2 Preprocessing

The datasets . in preprocessed to ensure consistency and compatibility with the model's architecture this may involves tasks like:

1. Manual Filtering:
   This involves Manually filtering out duplicate data points. Duplicate data can introduce biases and skew the learning process. By removing duplicates, you ensure that the model learns from a diverse and representative datasets. We also removed some unreadable signs and text copy from the web. The size of the genuine datasets is 4833 and fake datasets is 320.

2. Fake data Generation:
   We did generate the fake datasets done by the python code. The scale of the movie-rating of genuine datasets is between 0 to 10. The scale of the movie-rating of fake datasets is between 0 to 2 and 8 to 10. Since any attacker will always try to manuplate the genuine movie-rating datasets into the fake datasets in the own favourable who's fake rating could be very high or very low. That's why we did set the scale of movie-rating of fake datasets is between 0 to 2 and 8 to 10.

3. Train Test Split:
   The datasets was split in to 80/20 of training and testing. This way we used randomly 80 of our total datasets for training our and remaining datasets for testing our accuracy.

3.3 Algorithms

In this section, we will present the different models used for the purpose of this project.
The model was assessed as follows:

1. Random Forest Model
2. K-Nearest Neighbors Model
3. Voting Classifier

1. Random Forest Model: Random Forest Model is a commonly-used machine learning algorithm. Random Forest Model is developed by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Random Forest works by creating a bunch of decision trees, each trained on a different random subset of the data. For each tree, only a random set of features is considered when making decisions. When you want a prediction for a new data point, each tree gives its own prediction, and the final result is determined by a majority vote(for classification) or averaging(for regression) of all the individual tree predictions. This ensemble approach makes Random forest more accurate and less prone to over-fitting compared to a single decision tree.

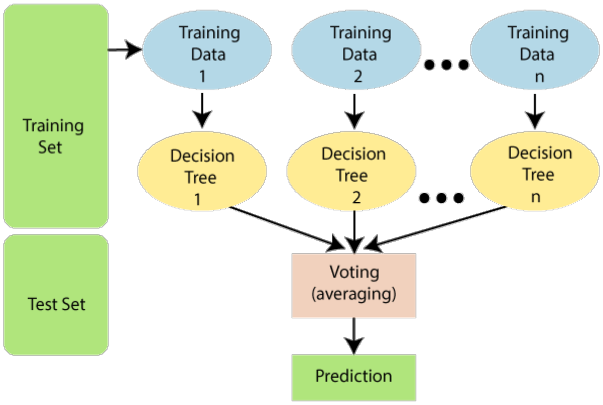

Fig. 1. Random Forest Model

2. K-Nearest Neighbors: The K-Nearest Neighbors algorithms is also known as KNN or k-NN algorithm. K-Nearest Neighbors classifier (KNN) used machine learning algorithm that is primarily used for its simplicity and case of implementation. Because of its simplicity and convenience of use, the machine learning algorithm K-Nearest Neighbors classifier (KNN) is utilized. No assumptions are necessary regarding the distribution of the fundamental data. In addition, it is adaptable to categorical and numeric data, rendering it a versatile option for a wide range of datasets utilized in classification and regression endeavors. This non-parametric technique bases its predictions on how similar the data points in a particular datasets are to one another. By comparison, K-NN is less susceptible to outliers than other algorithms. The K-NN algorithms locates the K nearest neighbors to a given data point using a distance metric, such as Euclidean distance, to determine its neighbors. By calculating the majority vote or average of the K neighbors, the class or value of the data point is subsequently ascertained. Using this method enables the algorithm to anticipate outcomes based on the local structure of the data and adjust to various patterns.

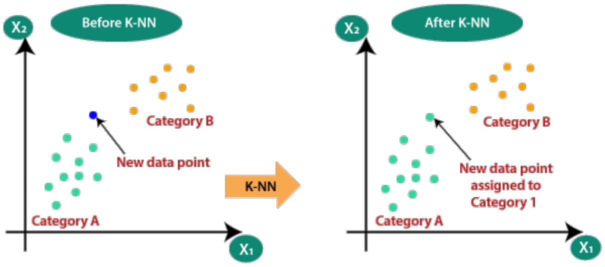

Fig. 2. K-Nearest Neighbors

3. Voting Classifier: A Voting Classifier is a machine learning model that trains on an ensemble of numerous models and predicts an output class based on their

highest probability of chosen class as the output. It simply aggregates the findings of each classifier passed into voting classifier and predicts the output class based on the highest majority of voting. In other simple words: The voting classifier is an ensemble learning method that combines several base model to produce the final optimum solution. The base Model can independently use different algorithms such as Random Forest Model, K-Nearest Neighbors Model, etc., to predict individual outputs.

There are two types of Voting Classifier.

1. Hard Voting
2. Soft Voting

1. Hard Voting: Hard voting is also known as majority voting. In Hard Voting, The predicted output class is a class with the highest majority of votes i.e the class which had the highest probability of being predicted by each of the classifiers.

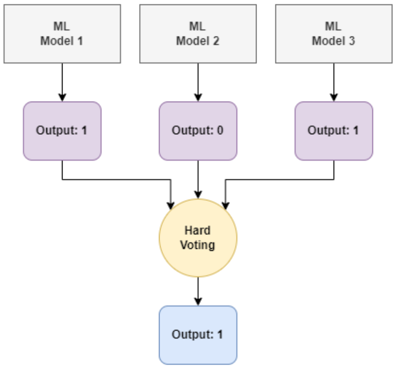

Fig. 3. Hard Voting

2. Soft Voting: In Soft Voting, The output class is the prediction based on the average of probability given to that class. In the end, the average of the possibilities of each class is calculated, and the final output is the class having the highest probability.

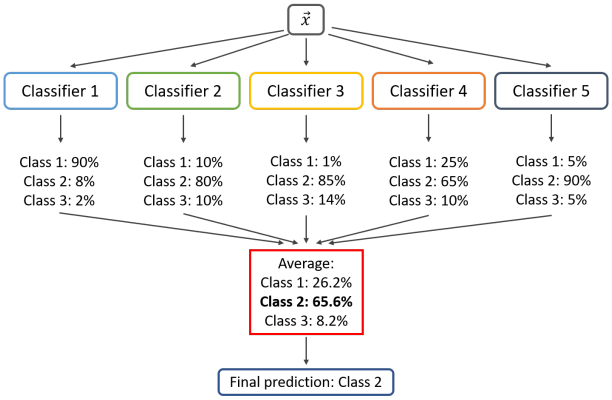

Fig. 4. Soft Voting

But In this Research Paper, We did use the Soft Voting Classifier.

4. Results

In this section, we evaluate and analyze the performance of various machine learning models.Accuracy was drastically increased after preprocessing the data. But to improve the accuracy further any model alone can't be sufficient so we tried to combine different Models. So in this projects, we did choose the different machine learning models such as: Random Forest Model, K-Nearest Neighbors Model, Voting Classifiers with highest accuracy scores. As you can see the accuracy scores of all models in table which is following below:

Accuracy Table

Model	Fake Data From Real Data	Fake Data From Fake Data
RF	0.70%	97.18%
KNN	12.77%	97.81%
Voting Regressor	3.27%	98.12%

4.1 Figures

On the figures below: Below results are generated on 4833 Genuine movie-rating datasets and 320 Fake movie-rating datasets.

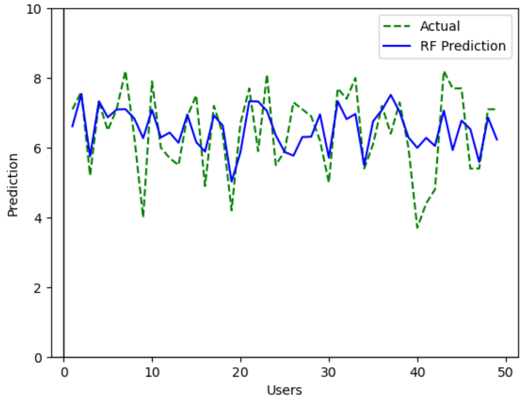

Fig. 5. Prediction Graph of Random Forest Model

5. Conclusions

In this Paper, we estimated the efficiency and the accuracy of machine learning algorithms in distinguishing between Genuine movie-rating and Fake movie-rating. We found that alone a model cannot be accurate and reliable so we merged two good performing models in order to achieve higher results. From accuracy table we can see that the Random forest model achieved the accuracy of 97.18% while the KNN model achieved the accuracy of 97.81% but voting classifier did achieved the high accuracy that is 98.12%.

Finally if we increase our datasets it will increase the reliability of the models. Further it can be used to detect Genuine movie-rating and Fake movie-rating. For Future we can

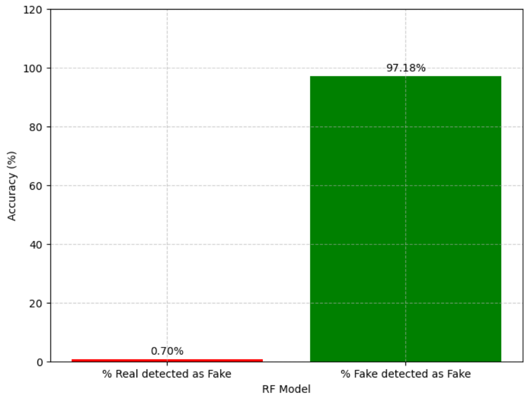

Fig. 6. Accuracy Graph of Random Forest Model

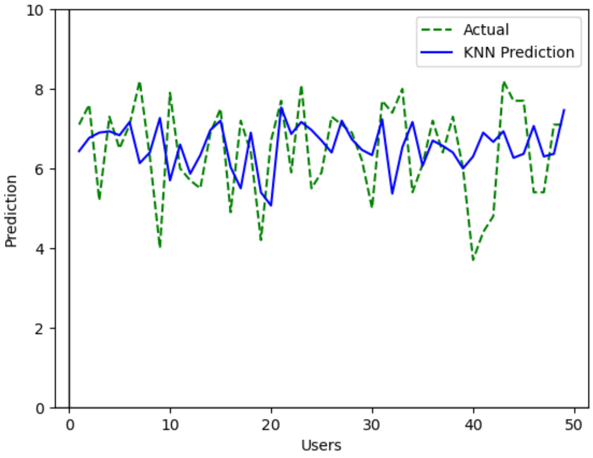

Fig. 7. Prediction Graph of K-Nearest Neighbors Model

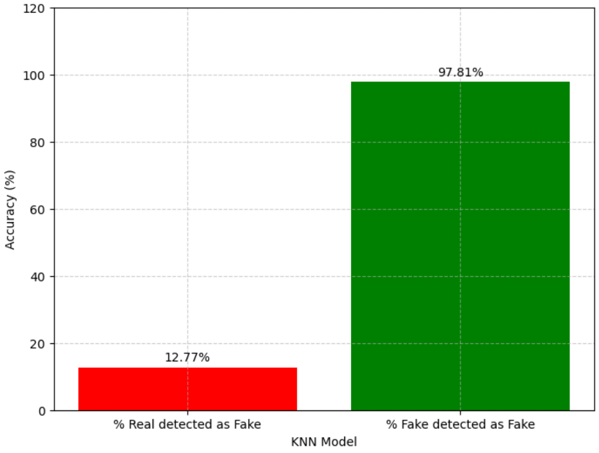

Fig. 8. Accuracy Graph of K-Nearest Neighbors Model

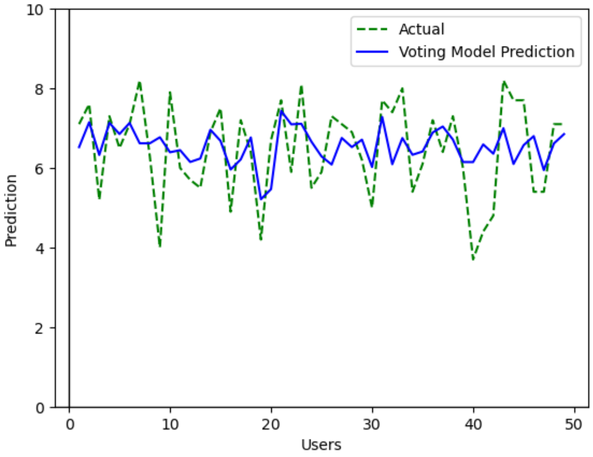

Fig. 9. Prediction Graph of Voting Classifiers

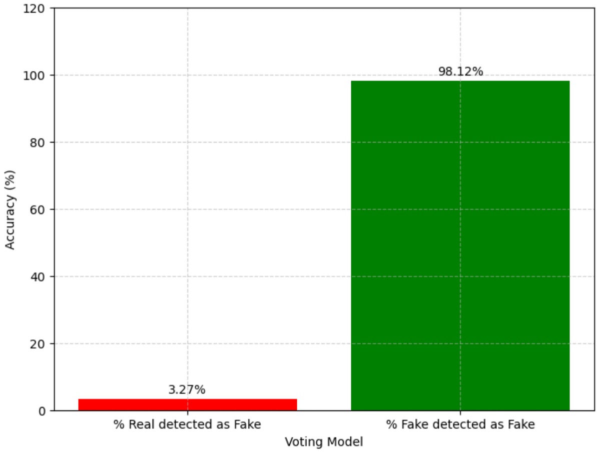

Fig. 10. Accuracy Graph of Voting Classifiers

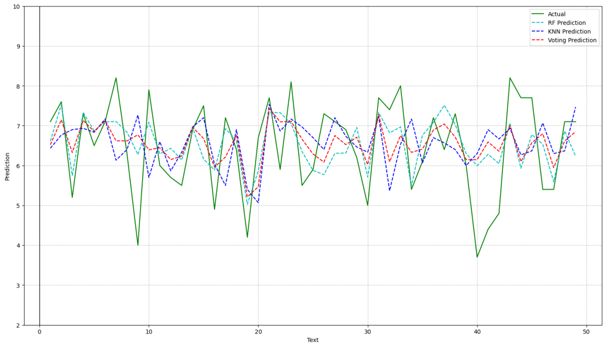

Fig. 11. Combine Prediction Graph of RF, KNN, Voting Classifiers Model

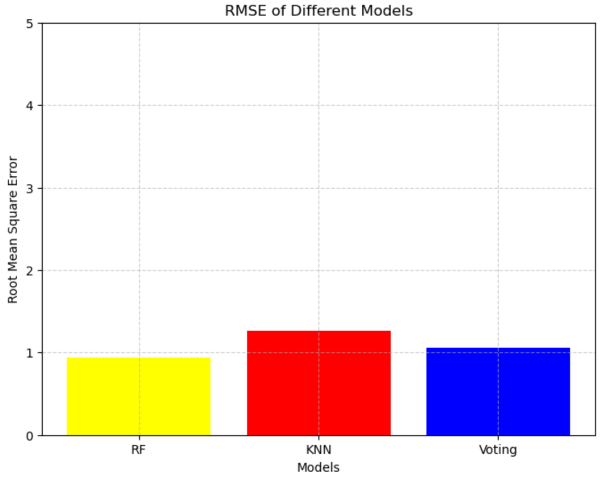

Fig. 12. Root Mean Square Error of RF, KNN, Voting Classifiers Model

make a website where user can upload their tittle of movie, Genres of movie, time duration of movie, rating of movie or any other relevant information that might impact a user's movie preferences in a specific situation or context and then our model will detect the movie-rating is Genuine or Fake. There are lot to do in this area, improvements can be done further.

6. Reference

1. Wei Zhou, Junhao Wen, Qiang Qu, Jun Zeng, and Tian Cheng. Shilling attack detection for recommender systems based on credibility of group users and rating time series. *PloS one*, 13(5):e0196533, 2018.
2. Tugba Turkoglu Kaya, Emre Yalcin, and Cihan Kaleli. A novel classification-based shilling attack detection approach for multi-criteria recommender systems. *Computational Intelligence*, 2023.
3. TheDevastator. Imdb movie ratings dataset. `https://www.kaggle.com/datasets/thedevastator/imdb-movie-ratings-dataset`, 2023. Accessed: January 17, 2023.