TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN CUỐI KỲ

HỌC PHẦN: KHOA HỌC DỮ LIỆU

ĐỀ TÀI:

DỰ ĐOÁN SỐ ĐIỂM IMDB CỦA PHIM

Nhóm	15
Võ Hữu Nam Trường	20.12
Nguyễn Công Anh	
Trần Xuân Nguyên	

ĐÀ NẴNG, 06/2023

TÓM TẮT

- Vấn đề: Trong thời gian gần đây, các phim mới được ra mắt ngày càng nhiều bên cạnh đó cũng có được rất nhiều sự quan tâm chú ý của khán giả. Ngoài nội dung của phim thì các thông tin khác về diễn viên, đạo diễn, nhà làm phim,... rất được nhiều người chú ý đến khi xem một bộ phim. Do đó, dự báo số điểm IMDB của phim là một hình thức để thông qua cách xem điểm số đánh giá về bộ phim, người xem có thể cân nhắc không xem những phim có đánh giá thấp để đỡ mất thời gian.
- Hướng giải quyết:

Nghiên cứu thu thập và mô tả dữ liệu, sử dụng mô hình Random Forest Regression, KNeighbors Regression với các kỹ thuật Feature engineering như:

  - Create New Features
  - Missing Value Imputation: Random, Median, Arbitrary Value Outliers
  - Categorical Data Encoding: LabelEncoder
  - Feature Scaling: RobustScaler
  - Feature Selection: SelectKBest
- Kết quả thu được:
  - Xây dựng thành công mô hình dự đoán điểm của phim.
  - Thử nghiệm và sử dụng các kĩ thuật feature engineering.

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Tự đánh giá
Trần Xuân Nguyên	Thu thập dữ liệu	Đã hoàn thành
Nguyễn Công Anh	Trích xuất đặc trưng	Đã hoàn thành
Võ Hữu Nam Trường	Mô hình hóa dữ liệu	Đã hoàn thành

MỤC LỤC

DANH MỤC HÌNH ẢNH

1. Giới thiệu

1.1. Đặt vấn đề

Trong thời gian gần đây, các phim mới được ra mắt ngày càng nhiều bên cạnh đó cũng có được rất nhiều sự quan tâm chú ý của khán giả. Ngoài nội dung của phim thì các thông tin khác về diễn viên, đạo diễn, nhà làm phim,... rất được nhiều người chú ý đến khi xem một bộ phim. Do đó, dự báo số điểm IMDB của phim là một hình thức để thông qua cách xem điểm số đánh giá về bộ phim, người xem có thể cân nhắc không xem những phim có đánh giá thấp để đỡ mất thời gian.

1.2. Giải pháp

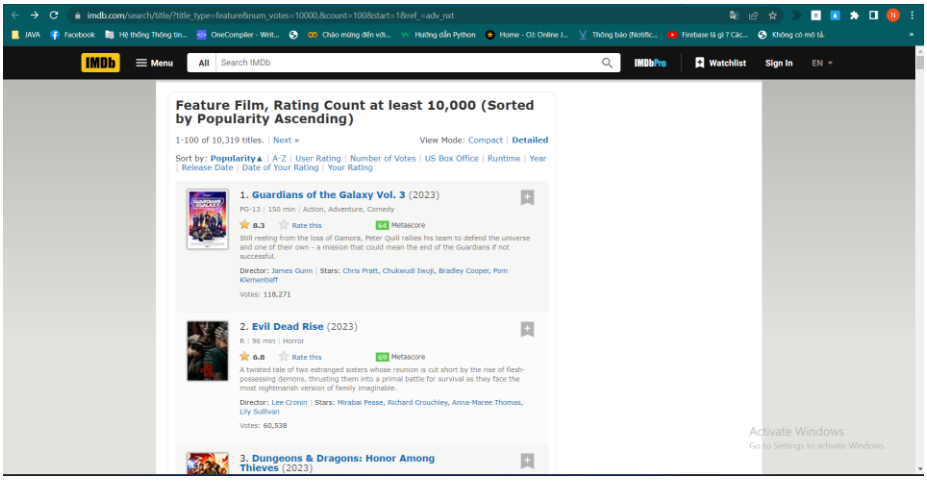
Nghiên cứu thu thập và mô tả dữ liệu, sử dụng mô hình Random Forest Regression, KNeighbors Regression với các kỹ thuật Feature engineering như:

- Create New Features
- Missing Value Imputation: Random, Median, Arbitrary Value Outliers
- Categorical Data Encoding: LabelEncoder
- Feature Scaling: RobustScaler
- Feature Selection: SelectKBest

2. Thu thập và mô tả dữ liệu

2.1. Thu thập dữ liệu

Dữ liệu được thu thập từ nguồn: https://www.imdb.com



Hình 1: Nguồn thu thập dữ liệu

Dữ liệu sau khi được request để thu thập dữ liệu từ nguồn trên. Bằng cách sử dụng BeautifulSoup, với BeautifulSoup là một thư viện Python dùng để lấy dữ liệu ra khỏi các file HTML và XML. Nó hoạt động cùng với các parser (trình phân tích cú pháp) cung cấp cho bạn các cách để điều hướng, tìm kiếm và chỉnh sửa trong parse tree (cây phân tích được tạo từ parser).

Để bóc tách hay parse một tài liệu HTML, ta cần cho nó vào hàm BeautifulSoup() – hàm này sẽ trả về một BeautifulSoup object. Bạn có thể truyền cho nó một file object đang mở hoặc một chuỗi html. Đầu tiên, tài liệu được chuyển đổi thành Unicode và các phần tử HTML được chuyển đổi sang ký tự Unicode, Soup được băm bằng parser tốt nhất hiện đang có.

Bản thân BeautifulSoup object đại diện cho toàn bộ tài liệu. Đối với hầu hết các mục đích, bạn có thể coi nó như một tag object. Điều này có nghĩa là nó hỗ trợ hầu hết các phương thức được mô tả trong điều hướng và tìm kiếm trong cây.

Ví dụ thu thập một mẫu ở dưới ta sẽ có được các dữ liệu sau:

Hình 2: Một mẫu để thu thập dữ liệu

2.2. Mô tả dữ liệu

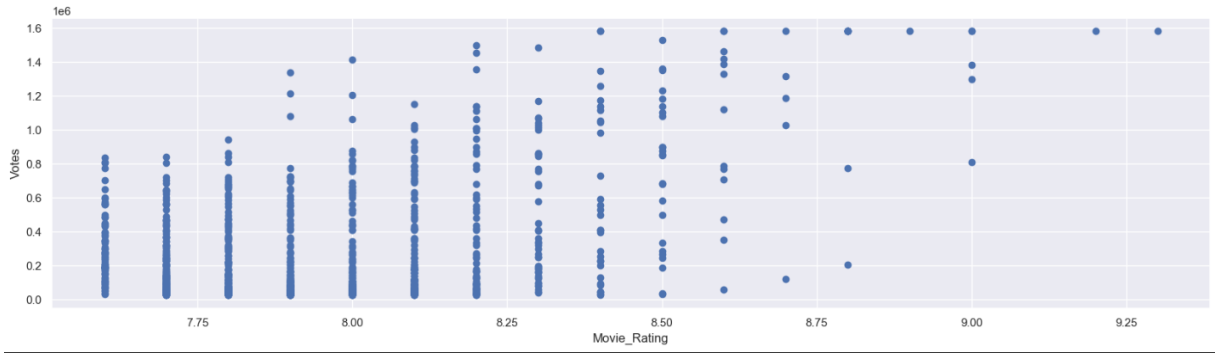
2.2.1. Số mẫu: 1000

2.2.2. Số đặc trưng của 1 mẫu

Tên đặc trưng	Kiểu dữ liệu	Số mẫu dữ liệu trống
Movie_name	string	0
Year_of_Release	int	0

C_Rating	string	16
Watch_Time_(min)	int	0
Genre	string	0
Movie_Rating	float	0
Metascore	float	156
Votes	int	0
Gross_($M)	float	186
Description	string	0

2.2.3. Mô tả trực quan các đặc trưng



Hình 3: Biểu đồ scatter cho thấy sự tương quan của Movie rating và lượng votes

Hình 4: Heatmap cho thấy sự tương quan giữa các đặc trưng với nhau

3. Trích xuất đặc trưng

3.1. Lựa chọn đặc trưng

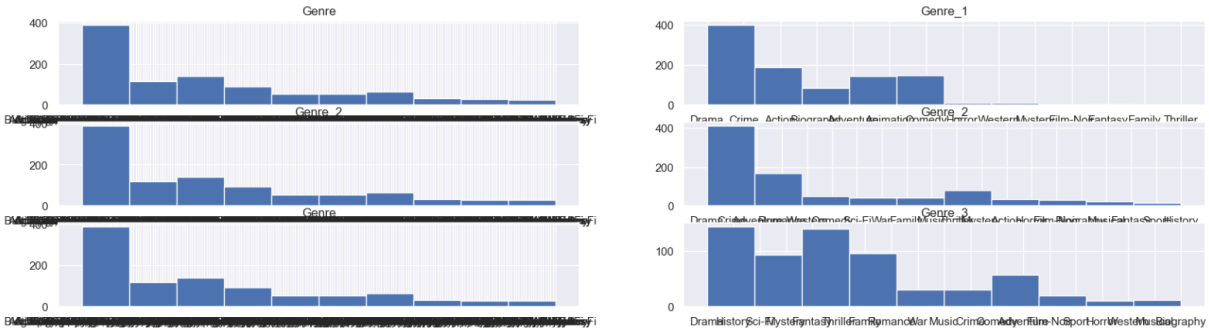
Lựa chọn đặc trưng dựa trên điểm số (scores) của các đặc trưng sử dụng SelectKBest. Phương pháp này hoạt động bằng cách chọn ra K đặc trưng có điểm số cao nhất, dựa trên một hàm đánh giá (score function) như Information Gain, Chi-square test, hoặc F-test. Các đặc trưng này được xem là quan trọng và có đóng góp cao hơn cho mô hình dự đoán.

3.2. Làm sạch và chuẩn hóa dữ liệu

3.2.1. Các kĩ thuật làm sạch dữ liệu

- Xóa cột thứ tự
- Xóa những bộ phim trùng lặp
- Xóa bộ phim có C_Rating là dữ liệu na

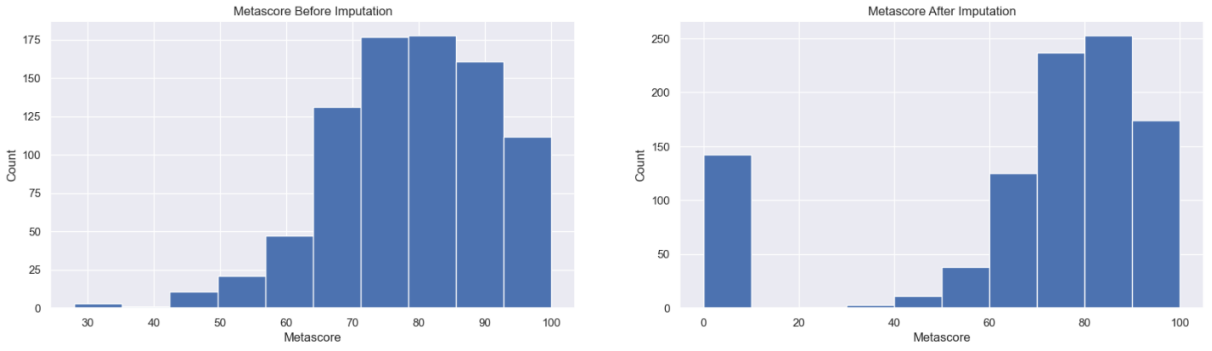
3.2.2. Tạo đặc trưng mới



Hình 5: Tạo các đặc trưng mới cho thể loại phim

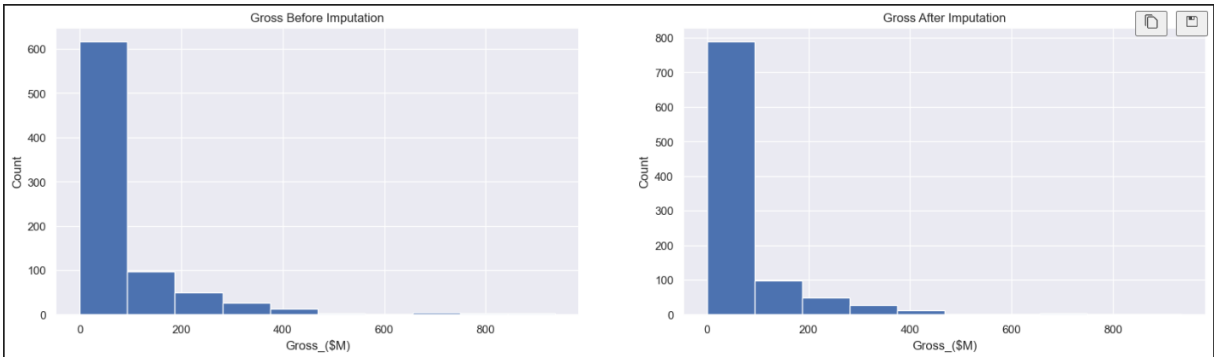
3.2.3. Xử lý dữ liệu trống

- Giá trị tùy ý:



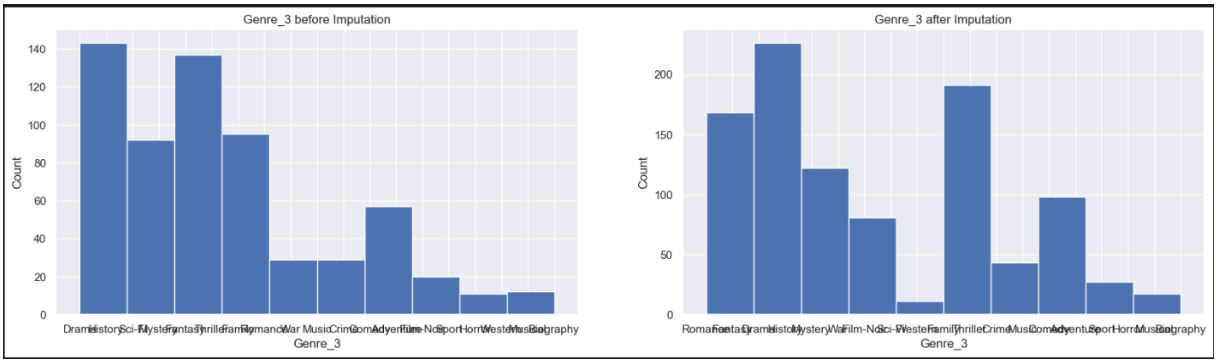
Hình 6: Xử lí dữ liệu trống của Metascore bằng giá trị 0

- Lấp dữ liệu trống bằng giá trị median:



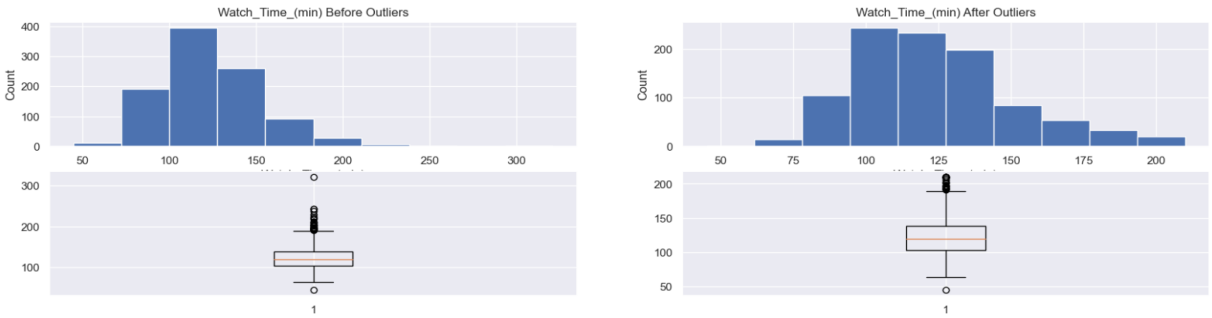
Hình 7: Xử lí dữ liệu trống của Gross bằng giá trị median

- Lấy dữ liệu trống bằng giá trị random:



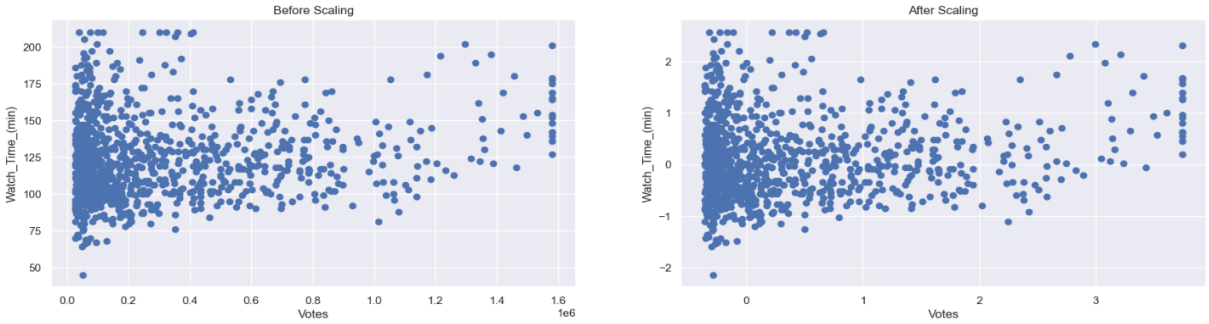
Hình 8: Điền vào dữ liệu trống của thể loại phim bằng các giá trị ngẫu nhiên

3.2.4. Xử lý ngoại lệ



Hình 9: Kết quả trước và sau khi xử lí ngoại lệ cho thời gian xem phim

3.2.5. Scaling



Hình 10: Trước và sau khi scaling dữ liệu sử dụng RobustScaler

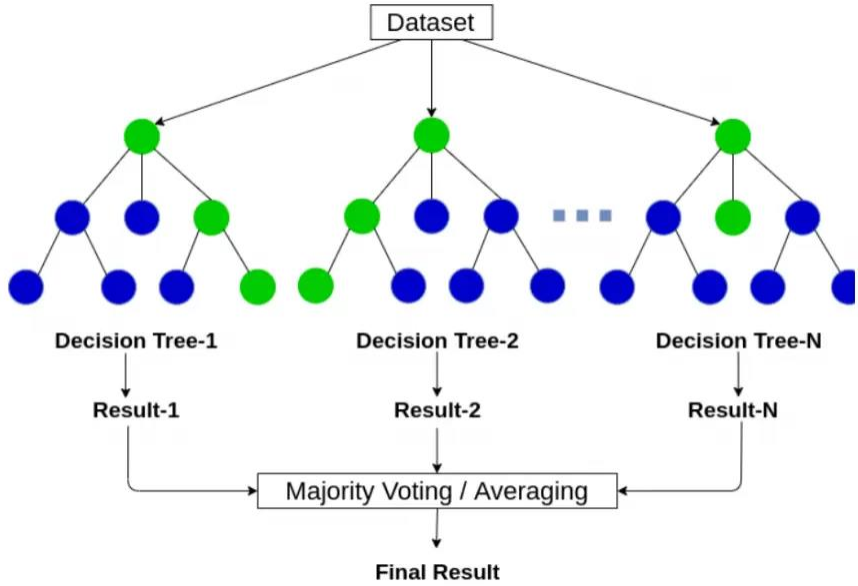
4. Mô hình hóa dữ liệu

4.1. Lựa chọn mô hình

Dựa vào việc mô tả dữ liệu và trích xuất đặc trưng, ta lựa chọn hai mô hình sau:

- Random Forest Regression
- K-Nearest Neighbors Regression

Random Forest: là một phương pháp thống kê mô hình hóa bằng máy (machine learning statistic) dùng để phục vụ các mục đích phân loại, tính hồi quy và các nhiệm vụ khác bằng cách xây dựng nhiều cây quyết định (Decision tree). Random Forest cho thấy hiệu quả hơn so với thuật toán phân loại thường được sử dụng vì có khả năng tìm ra thuộc tính nào quan trọng hơn so với những thuộc tính khác.

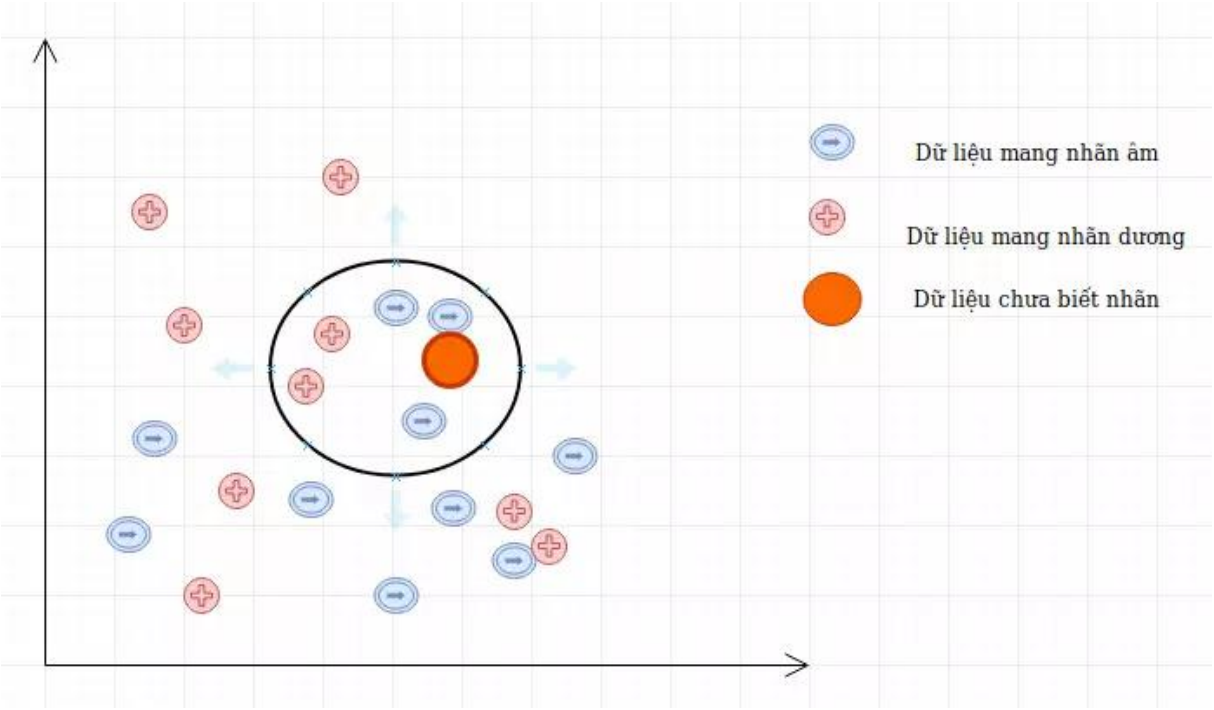


Hình 11: Mô hình thuật toán Random Forest

Trong mô hình RandomForestRegressor của thư viện sklearn có nhiều tham số, trong đó ta chọn 2 tham số chính là:

- **n_estimators, int**: The number of trees in the forest.
- **criterion, {"squared_error", "absolute_error", "friedman_mse", "poisson"}**: The function to measure the quality of a split.

K-Nearest Neighbors: là một trong những thuật toán supervised-learning đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong Machine Learning. Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. K-nearest neighbor có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression. KNN còn được gọi là một thuật toán Instance-based hay Memory-based learning.



Hình 12: Thuật toán K-Nearest Neighbors

Trong mô hình KNeighborsRegressor của thư viện sklearn có nhiều tham số, trong đó ta chọn 2 tham số chính là:

- **n_neighbors, int**: Number of neighbors to use by default for kneighbors queries.
- **weights, {'uniform', 'distance'}**: Weight function used in prediction.

4.2. Huấn luyện mô hình

4.2.1. Small Dataset

Ta chia dữ liệu thành các tập Huấn luyện/Xác thực/Kiểm thử theo tỉ lệ:

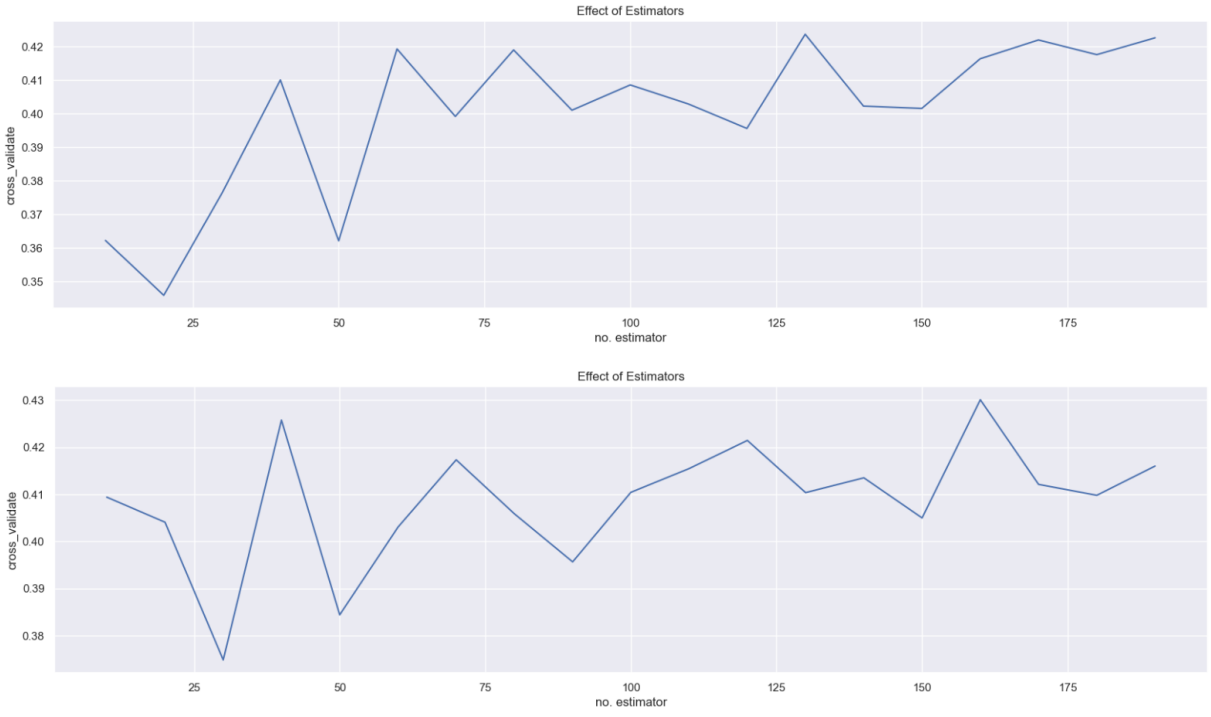
- 60% - train set,
- 20% - validation set,
- 20% - test set

Ta sẽ huấn luyện 2 mô hình sử dụng kĩ thuật **K-Fold Cross Validation** với các tham số trên:

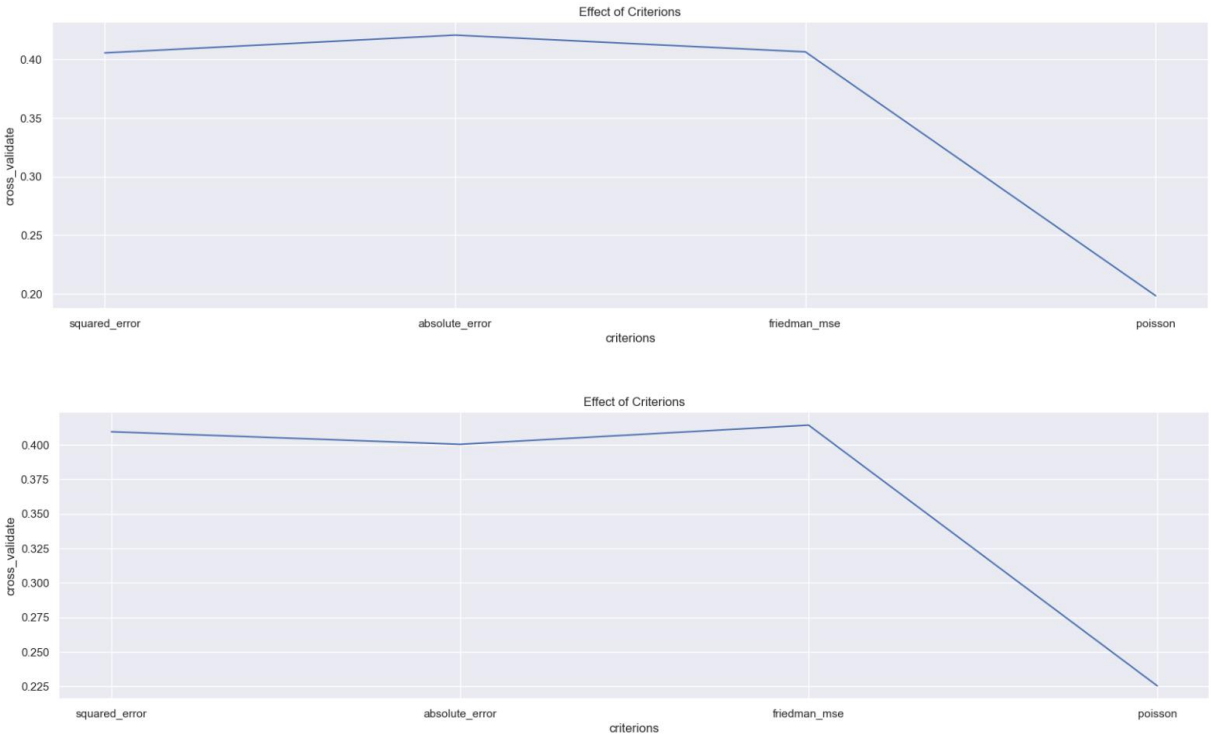
- **Random Forest:**
  - *n_estimators*: 10-200
  - *criterions*: squared_error, absolute_error, friedman_mse, poisson
- **K-Nearest Neighbors:**
  - *n_neighbors*: 1-20
  - *weights*: uniform, distance

Kết quả huấn luyện:

- ***Random Forest Regression***:
  - n_estimators: 80-160 (thay đổi liên tục)
  - criterion: squared_error/absolute_error/friedman_mse(thay đổi liên tục)

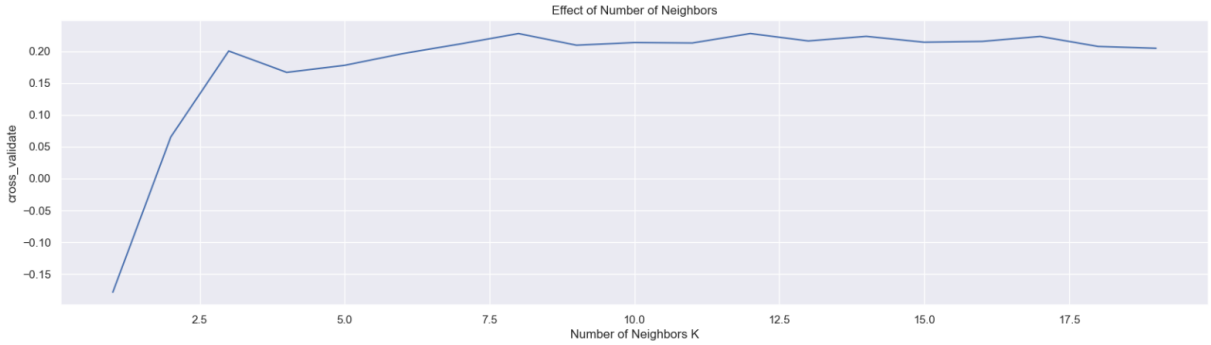


Hình 13: Đồ thị Cross Validation theo n_estimators

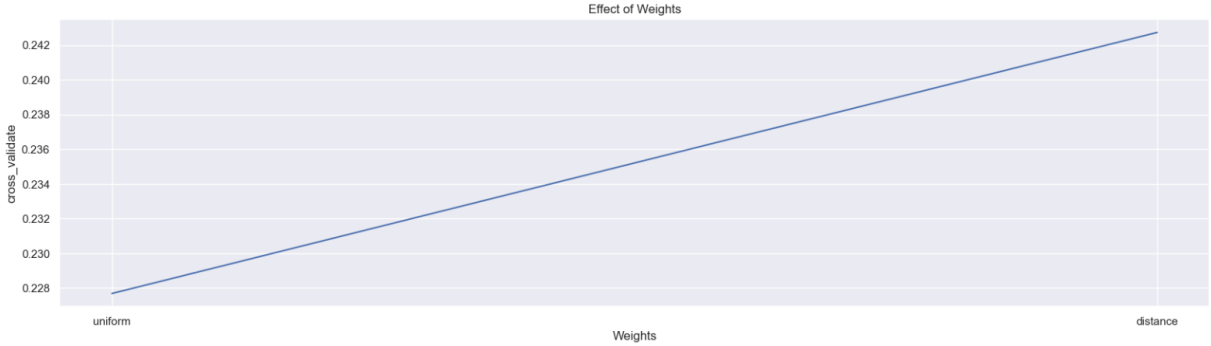


Hình 14: Đồ thị Cross Validation theo criterion

- ***K-Nearest Neighbors Regression****:*
  - n_neighbors: 8
  - weights: distance



Hình 15: Đồ thị Cross Validation theo n_neighbors



Hình 16: Đồ thị Cross Validation theo weights

4.2.2. Big Dataset

Ta chia dữ liệu thành các tập Huấn luyện/Xác thực/Kiểm thử theo tỉ lệ:

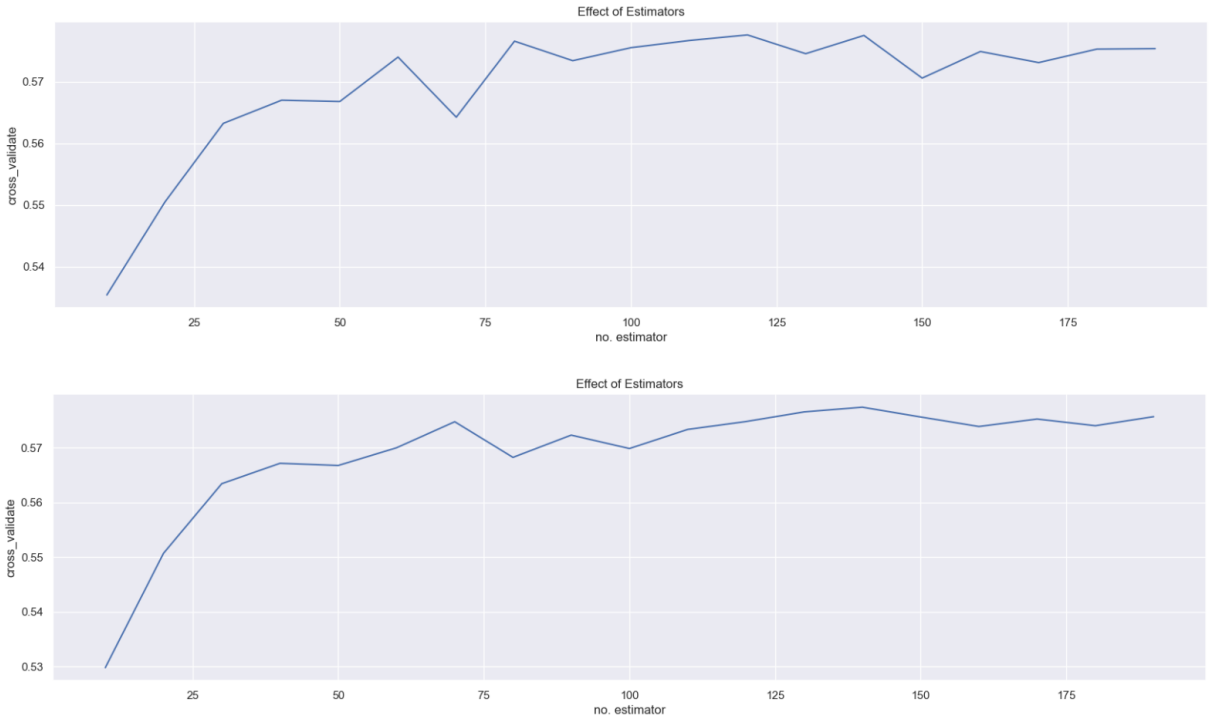
- 60% - train set,
- 20% - validation set,
- 20% - test set

Ta sẽ huấn luyện 2 mô hình sử dụng kĩ thuật **K-Fold Cross Validation** với các tham số trên:

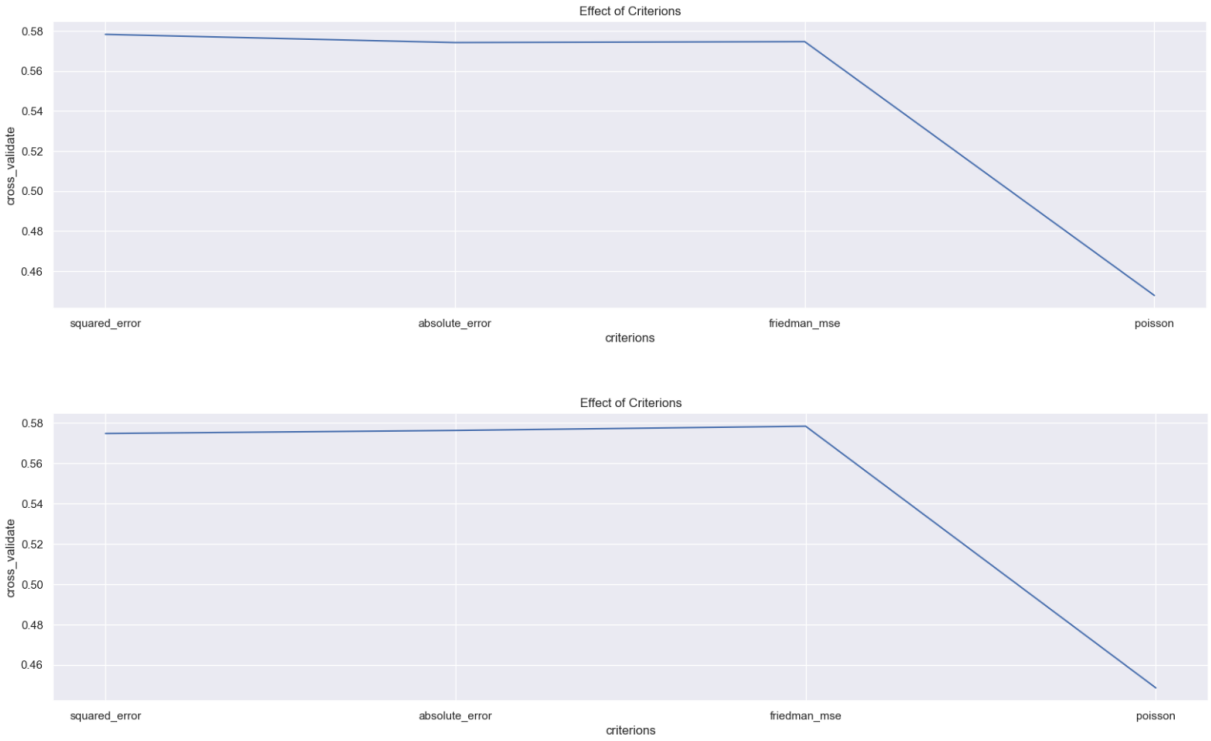
- **Random Forest:**
  - *n_estimators*: 10-200
  - *criterions*: squared_error, absolute_error, friedman_mse, poisson
- **K-Nearest Neighbors:**
  - *n_neighbors*: 1-20
  - *weights*: uniform, distance

Kết quả huấn luyện:

- ***Random Forest Regression***:
  - n_estimators: 100-190 (thay đổi liên tục)
  - criterion: squared_error/absolute_error/friedman_mse(thay đổi liên tục)

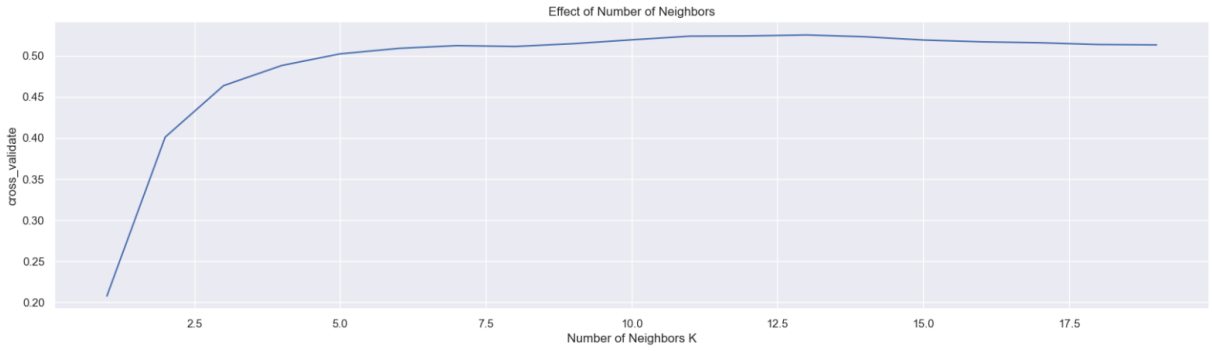


Hình 17: Đồ thị Cross Validation theo n_estimators

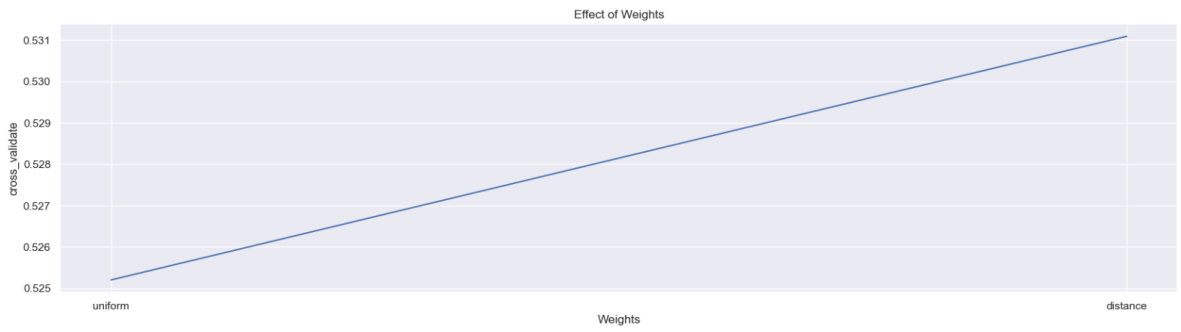


Hình 18: Đồ thị Cross Validation theo criterion

- ***K-Nearest Neighbors Regression****:*
  - n_neighbors: 13
  - weights: distance



Hình 19: Đồ thị Cross Validation theo n_neighbors



Hình 20: Đồ thị Cross Validation theo weights

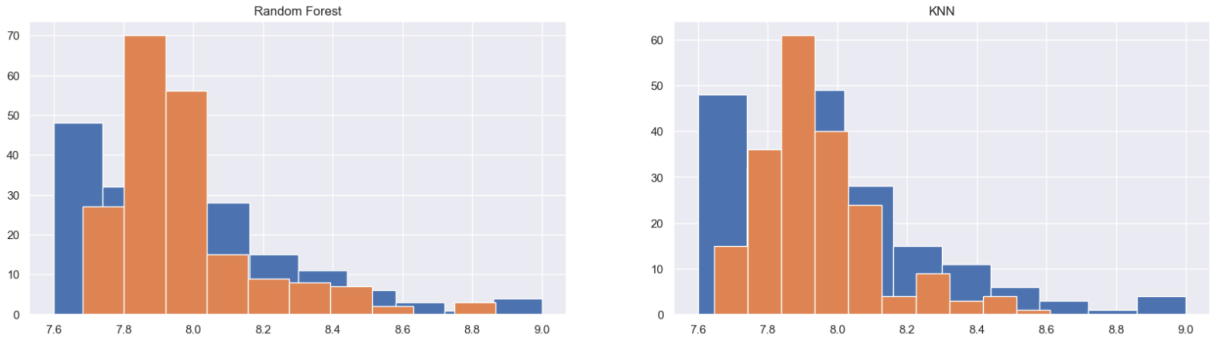
4.3. Đánh giá mô hình

4.3.1. Small Dataset

Ta có bảng đánh giá độ đo của các mô hình như sau:

	MAE	**RMSE**	**R2 score**
Random Forest	~0,1370	~0,1735	~62,02%
K-Nearest Neighbors	~0,1531	~0,1895	~54,66%

- Mô hình **K-Nearest Neighbors** có độ sai lệch MAE, RMSE cao hơn so với **Random Forest :** 0,0161 và 0,016
- Mô hình **K-Nearest Neighbors** có độ chính xác R2 score thấp hơn so với **Random Forest :** 7,36%



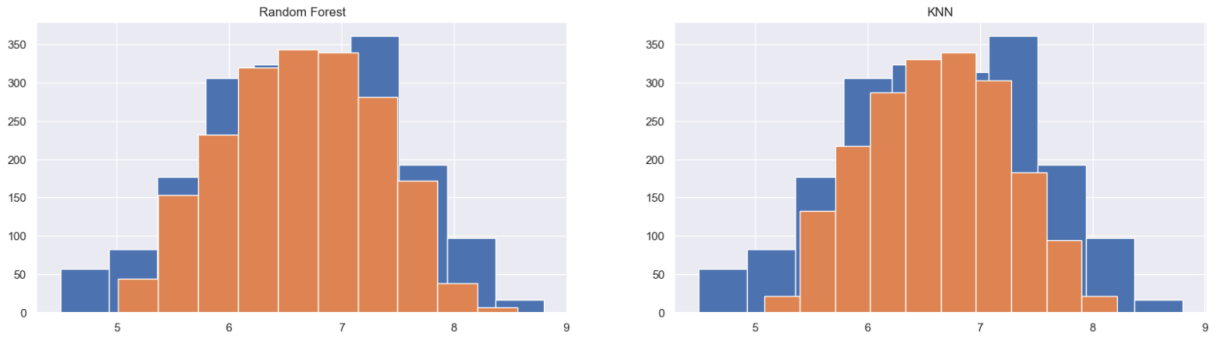
Hình 21: Kết quả dự đoán của 2 mô hình so với tập kiểm thử

4.3.2. Big Dataset

Ta có bảng đánh giá độ đo của các mô hình như sau:

	MAE	**RMSE**	**R2 score**
Random Forest	~0,3939	~0,5133	~63,77%
K-Nearest Neighbors	~0,4291	~0,5558	~57,52%

- Mô hình **K-Nearest Neighbors** có độ sai lệch MAE, RMSE cao hơn so với **Random Forest :** 0,0352 và 0,0447
- Mô hình **K-Nearest Neighbors** có độ chính xác R2 score thấp hơn so với **Random Forest :** 6.25%



Hình 22: Kết quả dự đoán của 2 mô hình so với tập kiểm thử

- So với kết quả của Small Dataset, các mô hình được huấn luyện bằng Big Dataset có độ sai lệch cao hơn đáng kể so với các mô hình được huấn luyện bằng Small Dataset
- Mô hình **Random Forest** có độ chính xác tăng so với trước: 1,75%
- Mô hình **K-Nearest Neighbors** có độ chính xác tăng so với trước: 2,86%

5. Kết luận

- Những việc đã làm:
  - Crawl dữ liệu với số lượng mẫu nhỏ (1000 mẫu) và số lượng mẫu lớn (10000 mẫu)
  - Sử dụng các kĩ thuật feature engineering (create new features, data cleaning, missing value imputation, label encoding, outliers, feature selection, feature scaling)
  - Xây dựng mô hình (sử dụng mô hình random forest và k nearest neighbor và so sánh 2 mô hình đó).
- Kết quả đạt được:
  - Xây dựng thành công mô hình dự đoán điểm của phim.
  - Thử nghiệm các kỹ thuật nâng cao độ chính xác của mô hình
- Hạn chế:
  - Độ chính xác của mô hình dự đoán còn thấp.
  - Chưa khai phá hoàn toàn dữ liệu được thu thập.

6. Tài liệu tham khảo

[1] Scraping Multiples pages of IMDB at a time to fetch top 1000 movies data - sivasahukar95

[2] IMDB Score Prediction for movies - SAURAV ANAND