Building user-based recommendation model for Amazon

DESCRIPTION

The dataset provided contains movie reviews given by Amazon customers. Reviews were given between May 1996 and July 2014.

**Analysis Task**  
- Exploratory Data Analysis:

* **Which movies have maximum views/ratings?**

movie\_views = movie\_views.sort\_values(by=["views"], ascending=False)

movie\_views[movie\_views.views==movie\_views.views.max()].moviename

* **What is the average rating for each movie? Define the top 5 movies with the maximum ratings.**

movie\_views\_max\_rating = movie\_views.sort\_values(by=["views","ratings"], ascending=[False, False])

movie\_views\_max\_rating.head()

* **Define the top 5 movies with the least audience.**

movie\_views\_max\_rating\_min\_audience = movie\_views.sort\_values(by=["ratings", "views"], ascending=[False, True])

movie\_views\_max\_rating\_min\_audience.head()

- Recommendation Model: Some of the movies hadn’t been watched and therefore, are not rated by the users. Netflix would like to take this as an opportunity and build a machine learning recommendation algorithm which provides the ratings for each of the users.

* **Divide the data into training and test data**

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(result.drop(['rating','total\_rating'],axis='columns'), result.rating, test\_size=0.2)

* **Build a recommendation model on training data**

from surprise import KNNWithMeans

# To use item-based cosine similarity

sim\_options = {

"name": "cosine",

"user\_based": False, # Compute similarities between items

}

algo = KNNWithMeans(sim\_options=sim\_options)

* **Make predictions on the test data**

trainingSet\_train = data.build\_full\_trainset()

algo.fit(trainingSet\_train)

prd = algo.predict(4067, "Movie140")

prd.est