Describe the movie recommendation learning problem by stating as precisely as possible the task, performance measure, and training experience

The task: Suggesting movies that a user will like. There are different approaches to this task. We could suggest movies just based on nearest neighbors of similar movies and hence make it a completely unsupervised system. We could also train a supervised model where we try to predict how a user might rate a certain movie. Let us for now assume we take a supervised approach and try to predict a users rating given specific movies features. (We then recommend the movies the user would give the highest rating) In addition to movie features (content-based filtering) we should also use features from other users (collaborative filtering), i.e. take into account what movies other users liked who liked a specific movie.

Performance Measure: The performance measure could be a regression metric like R^2 if we decide to go with the task of predicting a users rating on a movie. Hence we measure how far our prediction is from the rating the user actually gave to a certain movie.

Training Experience: We are likely to get very sparse matrices, as not all users rate all movies, hence reducing the sparsity of the data may be key to speed up training. To tackle this we can use dimensionality reduction techniques like PCA. It might also make sense to focus only on superusers and filter out users with too few ratings.

For each task below, specify what type of machine learning problem it is (i.e., supervised or unsupervised; classification or regressions or others). Explain your reasoning briefly in 1-2 sentences each.

Predict the stock market index

Supervised Regression. As stock market prices are timeseries data, one can treat future prices as the prediction labels, hence a supervised setting. Prices are continuous, hence regression.

Identify whether or not an alumni is going to donate to PKU

Supervised Binary Classification. We can train a classifier to predict the binary label if an alumni will donate.

Recommend online courses that are better taken together

Recommendation System – Can be supervised or unsupervised. Can be Regression. Recommendation algorithms can include clustering such as KNN, but often use filtering approaces like content / collaborative-based filtering, which search for similar content / users.

Segment customers based on social-demographic attributes

Unsupervised Clustering. We can use clustering algorithms like KNN to derive X groups from the customer data with similar attributes.

If we chose “Holiday” as the root of a decision tree, what would be the effects? Explain in terms of information gain.

Information Gain Formula

g(D, Holiday) = H(D) – H(D I Holiday)

Compute Entropy of Target Variable

H(D) = - 4/8 log_2 (4/8) + (-4/8 log_2 (4/8)) = -1/2log(1/2) – 1/2log(1/2) = ½ log(2) + ½ log(2) = 1

Compute Entropy of Target Variable if we know Holiday

H(D I Holiday) = 8/8 * (- 4/8 log_2 (4/8) + (-4/8 log_2 (4/8))) = 1

g(D, Holiday) = 1 – 1 = 0

The information gain of choosing Holiday as the root variable would be 0. It would not bring us any new information, since holiday is always F in our dataset. There would just be one node beneath our root tree. No decision would take place. We should just drop the H variable.

If “Holiday” is not proper as the root node, which attribute will you choose as the root node of the decision tree? Explain your reasons with necessary calculations.

H(D I Snowstorm) = 4/8 * (- 2/4 log_2 (2/4) + (-2/4 log_2 (2/4))) + 4/8 * (- 2/4 log_2 (2/4) + (-2/4 log_2 (2/4)) = 2 * (1/2 * (- 1/2 log_2 (1/2) + (-1/2 log_2 (1/2)))) = 1

g(D, Snowstorm) = 1 – 1 = 0

H(D I Long Distance) = 4/8 * (- 3/4 log_2 (3/4) + (-1/4 log_2 (1/4))) + 4/8 * (- 1/4 log_2 (1/4) + (-3/4 log_2 (3/4)) = 0.5 * (-0.75 log_2(0.75) + (-0.25 log_2(0.25) + 0.5 = 0.8112781244591328

In Python: 0.5 * (-0.75 * math.log(0.75,2) – 0.25 * math.log(0.25, 2)) + 0.5 * (– 0.25 * math.log(0.25, 2) -0.75 * math.log(0.75,2)

g(D, Long Distance) = 1 – 0.8112781244591328 = 0.18872187554086717

I will choose Long Distance as the attribute as it gives us an information gain of ~0.19.

Describe your complete decision tree in words (sample format: If there is a snowstorm, the flight will be canceled).

If the flight is long distance and if there is no snowstorm, the flight will happen.

If the flight is long distance and if there is snowstorm, the flight may be cancelled.

If the flight is no long distance and there is no snowstorm, the flight may happen.

If the flight is no long distance and there is snowstorm, the flight will be cancelled.

(Root: Long Distance; 2nd Node: Snowstorm)