Data Mining, Fall 2021
Final Exam

This exam is take-home, but is to be done INDIVIDUALLY, not in groups and not discussing with each other. You are free to use your book, the web, and your past assignments as you see fit.

Each of the following is a short-essay answer. Limit your answers to at most N words for each question, where N is specified for each question. Please type your answers directly into this word file and use the sample 12pt font size and double line spacing. Feel free to add tables/graphs as you see fit if that helps improve your answer. You do not need to run any code/experiments to answer these questions, but, if for some reason you want to you can.

Question 0:

I, ___Krista Miller________________________________, do declare that all work on this exam is my own and I have not discussed this with anyone.

Question 1:

The quality of data affects the usability of data mining results. Normalizing and data modification may be necessary to remove noise to develop accurate, meaningful results.

Association Rules are used to explain patterns in data from seemingly independent transactional data sets. Association rules use support, confidence, and lift measurements to identify the most interesting item sets. However, these rules may not be meaningful if the data isn't cleaned prior to running the algorithm. For example, exploring association rules from text becomes more useful once the text is converted to lower case and common words are dropped prior to running the algorithm. If the text were not converted to lower case, the Apriori algorithm would see words like "Dog" and "dog" as two separate entities, thereby diluting the association rule. Once all "Dog" occurrences are converted to "dog", it increases the frequency of the item occurring, thereby improving the value of the algorithm.

Decision trees build a set of decision rules, which function to predict an outcome from the input data. The approach builds a structure through a series of splits from the root node, passing through several decision nodes to arrive at the terminal leaf nodes. Each split partitions the input variable into feature regions, which are used for lower splits. One way to clean data to improve decision tree performance is through concept hierarchy, a data smoothing technique that bins the data into categories. For example, rather than looking at each age as a unique node, it may be more valuable to group ages as "20-30", "30-40", "40-50", and so on.

K-means algorithm attempts to split an anonymous dataset into a fixed number of clusters. It uses numerical data, is sensitive to outliers and noise, assumes a symmetric distribution of variables, and variables are on the same scale (generally an interval from [0.0, 1.0]). That way, this algorithm can consider all attributes equally. For example, standardizing height measured in cm and weight measured in kg converts the original measurements into 'unitless' variables, and eliminates the dependence on measurement units. Other common data preprocessing for k-means clustering is removing highly correlated variables, handling outliers, and managing data inconsistencies.

The **Naïve Bayes Classification** model is a classification technique with an assumption of independence among the predictor variables. It is calculated as $P(X|C_i)$ as the product of probabilities $P(x_1|C_i)$, $P(x_2|C_i)$, …. $P(x_n|C_i)$. However, if a categorical variable has a category in the test set that was not observed in the training data set, then the model will assign a "0" probability and will be unable to make a prediction. Therefore, a common data modification technique is Laplace correction, which adjusts data to solve the 'zero' probability problem by adding one to each count in a large database and not impacting the overall relative frequency of the class.

DBSCAN is a clustering algorithm. It uses distance and a minimum number of points per cluster to classify a point as an outlier. It is used to separate clusters of high density from clusters of low density. Unlike k-means clustering, DBSCAN does not require a number of clusters prior to running the model, and can learn clusters of arbitrary shape. For example, if a set of geographic data is measured in km, m, miles, and feet, it is necessary to standardize the distance measurement in order to develop an accurate DBSCAN model.

Question 2:

Algorithm	Data Use Case Example	Pros	Cons
Association Rules	Finding frequent, interesting itemsets	-Rules are easy to understand and communicate -Does not require labeled data	-Inefficient (Apriori)
Decision Tree Classification	-Predict whether a customer will purchase an item, given a variety of attributes	-Intuitive and easy to explain -Does not require data normalization -Creates a comprehensive analysis of consequences along each branch -Can handle numerical and categorical features	-A small change in the data can case a large change in the structure of the decision tree -Computationally more complex than other algorithms -Can become large and difficult to interpret without pruning -Cannot be used for continuous value prediction
Naïve Bayes Classification	Simple classification solutions (spam|not spam)	-Easy implementation -Fast computing (linear time complexity) -Resilient to noise and outliers	-Assumes that data features are independent, which could lead to bias
K-Means Clustering	Can be used to find groups that have not been labeled by data.	-Easy implementation -Results are easily interpretable -Fast computing (linear time complexity)	-Requires that the number of clusters are predetermined (it doesn't develop the 'optimal' amount of clusters) -Does not perform well with noisy data or outliers
DBSCAN Clustering	Can be used for recommendation algorithms by identifying similar groups.	-In contrast to K-means clustering, DBSCAN does not require a user to specify the number of clusters prior to implementing the algorithm. -It performs well with arbitrary shapes, rather than just spherical clusters. -DBSCAN is known to be resistant to data outliers and noise.	-Does not work well in high dimensional datasets -Because it can separate high density clusters from low density clusters, this algorithm struggles with clusters of similar density

Question 3:

A model with a 99.99% accuracy may sound like a perfect solution to a problem. However, accuracy alone is not always an adequate metric to evaluate a model and other metrics may be more appropriate or relevant to consider. Depending on the situation, we may want to optimize for recall or precision to reduce other consequences of a model performance.

For example, detecting a rare event (such as a very scarce disease) in a large dataset or population is an imbalanced classification problem. The category of non-diseased people represents the majority of the data points and greatly outnumbers those who do have the disease. So, the accuracy of a machine learning model correctly identifying data points may not be the best measure of performance. Identifying 999 healthy people and missing the actual disease 50% of the time is impactful. However, an accuracy score for an image classification model identifying clouds in the sky has less consequences if a raincloud is missed 50% of the time.

Recall is the ability of a model to find all the relevant cases within a dataset. Recall = true positives/(true positives + false negatives) = disease correctly identified/(disease correctly identified + disease incorrectly labeled as not disease). In other words, recall measures false negatives against true positives. A perfect recall score of 1.0 means that all relevant disease was detected, and there were no false negatives. Recall is also referred to as sensitivity or true positive rate. Generally, we want to minimize false negatives in disease detection. However, a test with perfect sensitivity may also have consequences. For example, it may be inefficient to run an algorithm on a radiology image for a rare disease if it takes a long time to generate results thereby hindering care for more common diseases.

Precision is the number of true positives/(number of true positives + false positives). Precision measures the rate of false positives. If precision is lower than accuracy, it means that false positives are a larger part of our error set. For disease detection, we generally want to improve precision and reducing false positives because we don't want to be treating people for disease that they do not have. Therefore, to holistically evaluate the effectiveness of a model, we must examine precision, recall, and accuracy. Depending on the problem and the consequences of false negatives or false positives will determine how we want to optimize these metrics.

Question 4:

A data cube is a multidimensional array of data values. For example, a company may want to summarize data by product, time period, and location. These three dimensions can be organized in a hierarchy. Time can be aggregated in days, weeks, months, quarters, or years. Location may be organized by city, state, and country. This allows for a user to slice, pivot, and drill down/drill up the attributes in order to analyze and summarize historical, archived data. Often, this is a relatively simple and quick computational query.

However, there are scenarios where we do not have an understanding of how attributes are related and may need other ways to discover patterns in datasets. A data cube can help generate aggregates (frequency counts), which is necessary for computing support and confidence for association rules. We can fetch attributes that are stored in a data cube for classification and clustering problems. Association rules and clustering belong to unsupervised machine learning techniques where the algorithm is not provided with any pre-assigned labels for the training data. A company can use these tools to understand features about their customers that may not be obvious in a data cube structure. Classification is helpful for situations where we want to identify the most important attributes in order to assign observations to categories. Classification can be helpful with several pattern prediction problems, such as image classification, speech recognition, or recommender systems. These models can have consequences if they are not optimized for accuracy, precision, and recall.

One trade-off in a machine learning algorithm is that it may take time to test, evaluate, and explain the model. Unsupervised machine learning algorithms on large datasets (such as the Apriori algorithm for association rules) is generally computationally expensive and may not result in insightful, actionable information. Classification algorithms can have real-world consequences (high false positive rate, high false negative rate) which can be impactful, and should be continuously monitored for performance.