Team Project Proposal
Data Minor

Members:
Sang Choi
Jose Cruz

As a team project, team Data Minor will be implementing three classification algorithms: Naive Bayes, Decision Tree, and K-nearest neighbors. The implementation of these algorithms will be coded in Python. The correctness and performances will be compared against Scikit-learn library implementations. Namespaces of these Scikit-learn implementations are as follows: sklearn.naive_bayes, sklearn.tree, and sklearn.cluster.neighbors.KNeighborsClassifier.

The team will use git to collaborate and Anaconda for python package management. We will use one of Kaggle datasets that have various attribute types for validations and performance testing. Datasets will be kept small for simplicity sake. Testing will require use of various open-source python libraries. timeit library will be used to compare the average execution time and memory-profiler library will be used to compare the memory usage. Each algorithm will be validated using 10-fold cross validation and standard classification metrics such as accuracy, precision, and recall. The analysis will be focused on ensuring that there's no sampling bias and overfitting.

The team plans to start the project as soon as possible after the proposal. Each team member will each take an algorithm and have the other person validate the results once the development is finished. We will finish the rough draft implementations by November 1. From there, we will take a week to create a simple user interface for a user to enter in the location of the dataset to run classifiers with. Additionally, we will perform validations to ensure correctness of each algorithm and do performance tuning to minimize execution time and memory usage. Afterwards, we will work on the technical writeup summarizing our work until the deadline.

Below are brief descriptions for each algorithms:

Naive Bayes: The Naive Bayes theorem is a classification algorithm used for binary or multiclass classifications. It provides a way to calculate the probability of a piece of data belonging to a given class. It is naive in the sense that the calculations of the probabilities for each class are simplified to make their results tractable. They are assumed to be conditionally independent given the class value. It is as follows:

$$P\left(class\,|data\right) = \frac{(P(class|data)\cdot P(class))}{P(data)}$$

Decision Tree: A decision tree is a flow-chart like tree structure where an internal node represents a feature, the branches represent a decision rule, and each leaf represents the outcome. This algorithm is easy to understand and interpret by their visualization of flowchart diagrams. To make a tree:

1. Select your best attribute to split the data.
2. Make that attribute a decision node and break the dataset into smaller subsets.
3. Build the tree by repeating the process recursively for each child until one of the following conditions match:
    a. All tuples belong to the same attribute value

b. There are no more remaining attributes
c. There are no more instances

K-Nearest Neighbor Classifier: This is a type of classification algorithm used to cluster neighbors together to a predefined K value. Vectors in a multidimensional feature space are used for the training data. K is given by the user, along with a test point assigning a label to it its most frequent among the samples. With that, distances are then measured to find closest to the label.