CS 422/622 Project 2: K-NN, K-Means, Perceptron

Due 10/24 @ 11:59PM

Logistics: This project includes only one file, Iris.csv. This data will be used to develop and generate solutions for all three portions of this project. The expectation is to implement code for all three problems from scratch, and to answer the accompanying write-ups/discussions. The written portion of the project should not use any generative AI, and is required to be done on your own. Due to only one dataset being provided, you can hardcode the name into your code for easier testing, but please do not provide it as an absolute data path (ex. /home/user/project/exe) as it will need to be changed when running on our end for grading. For each problem, please keep the solution code in separate python files, so one for each portion of this project. The write-up can be done in a single file. You are only allowed the numpy, pandas, and matplotlib libraries for this project.

Deliverables: You should submit a single ZIP file, containing your project code (knn.py, kmeans.py, perceptron.py), the data file (Iris.csv) and your writeup (PDF). Grad students are also expected to include their LaTeX source files (∗.tex). Your zip file should be named lastname1_project1.zip. For example, a zip file for Sara Smith would be smith_project1.zip. Your code should run without errors on the ECC linux machines. If your code does not run for a particular problem, you will lose 50% on that problem.

1 K-Nearest Neighbor (20 Points)

File name: knn.py

Code Implementation (10 Points)

```python
def knn(k, X, y, x_test):
    ...
    return accuracy
```

Implement the K-Nearest Neighbor algorithm in python with the above function as the algorithm template. K refers to the value of amount of nearest neighbors to train with, while X and y refer to the data samples and their labels to train the model on. x_test should accept some other data samples to test the performance of the model. For grading, your code will be tested against a subset of the iris dataset for different values of k. The Iris dataset is provided for testing. The function should return the accuracy of the model, which should be output to the terminal.

Write-Up (10 points)
Give a brief description of the algorithm. Explain what the algorithm's strengths and shortcomings are, and describe why K = 1 is not a useful parameter for training.

2 K-Means Clustering (40 Points)

File name: kmeans.py

Code Implementation (20 Points)

```python
def kmeans(k, X):
    ...
    return clusters

def plot(clusters):
    ...
```

Implement the K-Means Clustering algorithm in python. You should implement the two functions listed. K refers to the number of random points to act as cluster centers. X refers to the data being passed (where the data only consists of 2 features of your choosing). The function should return the generated clusters. The returned clusters should then be passed to the plot function, which will generate plots using matplotlib. Make sure to color code your clusters for visibility. The plot function does not need a return. You will be graded on this portion based off the graphs you generate. The Iris dataset is provided for testing. You will want to output multiple plots where the x and y axes use different combinations of the features provided in the dataset. It is expected to try and use multiple values of K to find which parameter produces the most useful plots.

Write-Up (20 points)
Give a brief description of the algorithm. Explain what the algorithm's strengths and shortcomings are. Using the generated plots, perform some preliminary data analysis and describe your conclusions here in this write-up. For example, based on the generated clusters what conclusions can be made about the relationship between the features and the class labels? How varied can we say the dataset is? How much overlap is there between different classes? The next section is about implementing a perceptron, so which two classes appear to be the most linearly separable based on the clusters?

3 Perceptron (40 Points)

File name: perceptron.py

Code Implementation (20 Points)

```
def perceptron(X, y, x_test):
    ...
    return accuracy
```

Implement the Perceptron algorithm in python. The above function should accept data samples (X), labels (y), and a smaller subset of data samples for testing the performance of the model. The function should return the accuracy of the model. The Iris dataset is provided for testing. Using the clusters from the K-means portion of this project, pick only two of the classes to work with (perceptron's can only do binary classification) to implement your solution. The terminal should output the accuracy of the perceptron. You will be graded based off of a smaller subset of the Iris dataset passed into the x_test parameter.

Write-Up (20 points)
Give a brief description of the algorithm. Explain what the algorithm's strengths and shortcomings are. Describe why data needs to be linearly separable, or close to in order for this implementation of a Perceptron to work.

Grad Students — Algorithm Design (10 points)
As noted, a Perceptron only works on binary classification problems. How can we modify or expand the Perceptron to work with multiclass problems. What do these modifications/expansions start to look like in relation to other Machine Learning algorithms?

Grad Students — Algorithm Implementation (10 points)

```
def multiclass_perceptron(X, y, x_test):
    ...
    return accuracy
```

Implement a solution for multiclass classification using Perceptrons and test it against the entire Iris dataset. As a hint, you will need to perform classification for each pair of classes (class 1 vs class 2, class 1 vs class 3, etc.).

Grading: Your code will be ran in the latest Ubuntu environment as provided. If the code fails to compile, it will be a 50% penalty. Your code will be ran as is with the Iris.csv data as provided. The functions provided here in the instructions are simply where the algorithm is intended to be implemented. You are free to implement whatever you need outside of the functions to get the code working. 422 students are graded on a 100 point scale. 622 students are graded on a 120 point scale.

Grad Students — LaTeX Files:
Grad students are expected to write their report in LaTeX and include the source .tex file with their pdf upload. If you are in need of a resource to write LaTeX files, overleaf.com is an excellent free resource available.