Project 2: Feature Selection with Nearest Neighbor

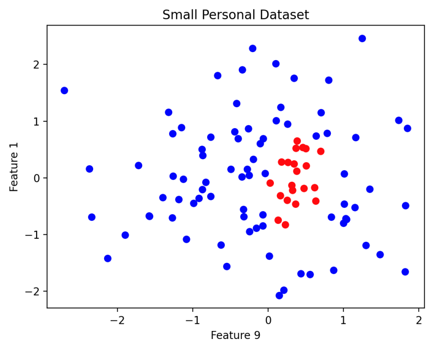
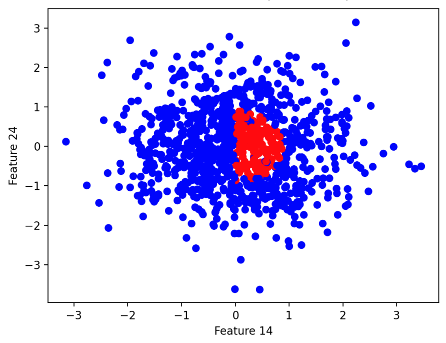
Student Name1: Kartik Gulia SID: 862258443 Lecture Session: 002

Student Name2: Rayyan Zaid SID: 862291205 Lecture Session: 002

Student Name3: Aishani Patalay SID: 862289220 Lecture Session: Seccion 002

Student Name4: Cris Liao SID: 862396679 Lecture Session: 002

Solution: Group 31

Dataset	Best Feature Set	Accuracy
Small Number: <CS170_Spring_2023_Small_data__31.txt>	Forward Selection = {9,1} 	0.98
	Backward Elimination = {1, 5, 7, 9}	0.89
	Custom Algorithm = Not implemented	N/A
Large Number: <CS170_Spring_2023_Large_data__31.txt>	Forward Selection = {14,24} 	0.958

	Backward Elimination = {1, 2, 4, 14, 15, 16, 18, 19, 21, 22, 23, 29, 31, 33}	0.807
	Custom Algorithm = N/A	N/A

------------------------------<Begin Report>---------------------------------

In completing this project, I consulted following resources:

https://docs.python.org/3/tutorial/inputoutput.html

Class Notes

Contribution of each student in the group:

Kartik: Set up the github repo, did forward selection, and report.
Rayyan: Did the dataset validation (Validator Class) , the classification model (Classification Class), and the datasetActions.py which has various actions performed on the dataset.
Chris: Worked on Backward Elimination.
Ash: Worked on Backward Elimination and report

I. Introduction

The project assigned to us was creating a Feature Selection with Nearest Neighbor. This was divided into three parts and each stage can be seen by the github repo. The first part was creating two greedy search algorithms: forward selection and backwards elimination. This was where we input the number of features, and test out different combinations depending on the algorithm, and using a random accuracy function we find the accuracy. For the second part we actually implemented an evaluation function (leave-one-out validation) and the NN classifier. In the third part we combined parts one and two where we used the greedy search algorithms on real data with the actual evaluation function. We thoroughly tested this using both the datasets initially given as well as our own personal datasets.

II. Challenges

When creating the forward selection a challenge I came across was making sure the greedy algorithm would expand correctly even if that level didn't have a better accuracy then a previous level. I was able to overcome this by creating two sets, one for the best and one for the current as well as a checker to compare and creating checkers as well, to see if accuracy is the best or it decreased.

III. Code Design

Github repo: https://github.com/kartikgulia/Feature-Selection-with-Nearest-Neighbor

main.py:

- Introduces our group and project
- Opens file
- Enter total number of features and which algorithm you want to run
- Depending on the algorithm chosen goes to said functions in other files

forwardSelection.py:

- Starts with null, gets accuracy, then tests out the initial level of combinations, based on the best accuracy of that level it expands, and repeats this process until all combinations have been tested and outputs which combination has the highest accuracy.

backwardElim.py:

- Opposite of forward selection, starts with all of the features in a set, takes one out and sees the accuracies of each possibility, reduces the one with the best accuracy, and repeats until you get to null. Then output the combination with the overall best accuracy.

randAccuracy.py:

- Just for part 1, gets a random number between 1 and 100

Classifier.py:

- Trains the Classifier with the input data from the text files. Training the nearest neighbor classifier is just taking all the input data from the file and storing it in a data structure so that it can be accessed efficiently when comparing new testing points. In our case, we used a dictionary (hashmap) to store a training instance : [subsetOfFeatures] key:value pairing. This allowed for organized structure and lookup when testing new points.
- Tests the Classifier and returns the predicted class label. We do this by going through our map and comparing every single point in the map to the current point we are testing by taking the euclidean distance of each feature.

datasetActions.py

- Reads the dataset and stores it into a 2D float array so that the data can be read by Python
- Filters the dataset by the specific feature subset that is chosen. This helps during validation and makes the whole process more organized.

Validator.py:

- Uses Classifier.py and datasetActions.py to calculate the accuracy
- We go through each row of the dataset and leave that row out. During this, we have a counter that counts the number of correct predictions by the model. In the FS and BE searches, we compare the accuracy generated by each subset with the best accuracy so far to determine which subset is the best.

IV. Dataset details

The General Small Dataset:10 features, 100 instances

The General Large Dataset: 40 features, 1000 instances

Your Personal Small Dataset: 5 features, 40 instances

Your Personal Large Dataset: 20 features, 500 instances

V. Algorithms

1. Forward Selection
    - Starts with null, gets accuracy, then tests out the initial level of combinations, based on the best accuracy of that level it expands, and repeats this process until all combinations have been tested and outputs which combination has the highest accuracy.
2. Backward Elimination
    - Opposite of forward selection, starts with all of the features in a set, takes one out and sees the accuracies of each possibility, reduces the one with the best accuracy, and repeats until you get to null. Then output the combination with the overall best accuracy.

VI. Analysis

Experiment 1:

1. Comparing Forward Selection vs Backward Elimination.
    - The forward selection has higher accuracy than backward elimination on a small dataset. However, the backward elimination has higher accuracy than forward selection on a large dataset.
2. Compare accuracy with no feature selection vs with feature selection.
    - We use no feature selection, the accuracy is not accurate. However, when we use the feature selection, we compare different subset of features in order to get the highest accuracy, and the result is more accurate.

3. Compare feature set and accuracy for forward selection vs backward elimination.
    - The forward selection has an increasing current best subset since it adds 1 feature while the loop is running. The backward elimination has a decreasing current best subset since it reduces 1 feature while the loop is running.
    - By using a small dataset, the forward selection has higher accuracy than the backward elimination. However, by using a large dataset, the forward selection has lower accuracy than the backward elimination.
4. Pros and Cons of Forwards Elimination and Backward ELimination

    Forward selection:

    Pros: Can be good with dealing with large numbers of features and is simple to implement, which can pick out the features that give us more insight when analyzing data.

    Cons: Some of the negative aspects of forward selection is that it is more susceptible to overfitting. Another disadvantage is that it may not always find the most optimal feature set.

    Backwards Elimination:

    Pros: One advantage is that it can be more effective at finding the optimal features compared to forward selection. Compared to the forward selection it can reduce the problem of overfitting by taking away features that are irrelevant, which gives a good evaluation of each feature.

    Cons: Some of the disadvantages of using backward elimination is that it can be computationally expensive, especially if we are using a large number of features and need precise stopping criteria to prevent overfitting or underfitting.

Dataset	Best Feature Set	Accuracy
Small Dataset	Forward Selection = {3,5}	
	Backward Elimination =	
	Custom Algorithm = Not implemented	N/A
Large Dataset	Forward Selection = {1,27}	
	Backward Elimination =	
	Custom Algorithm = N/A	N/A

Experiment 2: Effect of normalization

- Using the unnormalized data will get the more inaccurate distance calculations. The nearest neighbor algorithm measures distances of points in order to make a prediction and the data features having different scales/units would make the predictions less accurate. This can be because larger scales and ranges are more favored, which may overlook other features leading to a biased featured selection. The normalized data will have a more accurate accuracy due to the fact of having the same scaling factor. It would make the features equally important to one another and would avoid the issue of having one feature dominate the other because the standard deviation would not be large.

Compare accuracy when using normalized vs unnormalized data.

Experiment 3: Effect of number neighbors (k)

- The smaller values of k might lead to the overfitting. However, the large values of k might lead to underfitting. We used the validator file to use cross-validation to find a suitable value for k so overfitting or underfitting would not be an issue

Plot accuracy vs increasing values of k and examine the trend.

VII. Conclusion

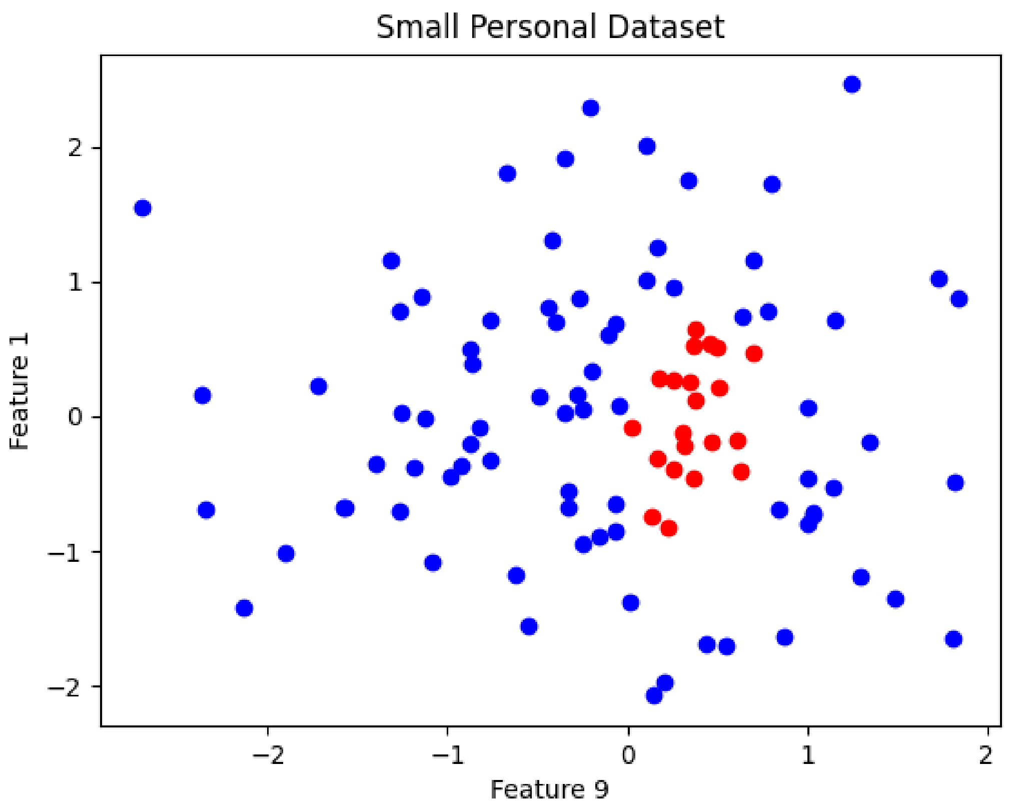
Based on our analysis, the backward elimination gets higher accuracy when using a large dataset and more features. The forward selection should get higher accuracy when using a small dataset and less features. It is clear that these two algorithms each have unique traits in general.The distinction between these two fundamental algorithms is in the initial instructions. We set the initial subset to be an empty set, and keep adding 1 feature to the current best subset for forward selection. We set the initial subset to be a full set of features, and keep reducing 1 feature to the current best subset for backward elimination. I believe that in addition to the algorithm's quality, we should take the problem's frequent demand into account. Impose that we think about what we need more. For instance, which is more important to us: accuracy or time? Different algorithms are chosen as a result of different answers.

Overall, the experiments highlighted the importance of feature selection, normalization, and choosing an appropriate number of neighbors in improving the accuracy of machine learning models. The specific techniques and considerations discussed for each experiment can be applied in various scenarios to optimize model performance.

VIII. Trace of your small dataset

Label 1 - Red

Label 2- Blue


Small Personal Dataset
Feature 1
Feature 9