Midterm Report

**Project Details:**

* Classifiers used:
  + Logistic Regression
  + K-Nearest Neighbor, k = 5, Euclidean distance
  + Support Vector Machine, Gamma = 0.001, c = 1, kernel = linear
* Dataset used: iris flower dataset

**Classifier Code:**

* Logistic Regression:
  + 24 def logistic\_regression (x\_train, y\_train, x\_test, y\_test):  
     25 logreg\_model = LogisticRegression()  
     26 logreg\_model.fit(x\_train,y\_train)  
     27 y\_pred=logreg\_model.predict(x\_test)  
     28 print("\*\*\*\*\*Logistic Regression\*\*\*\*\*")  
     29 print(classification\_report(y\_test, y\_pred))
* K-Nearest Neighbor:
  + 32 def k\_nearest\_neighbor (x\_train, y\_train, x\_test, y\_test):  
     33 le = LabelEncoder()  
     34 labels = le.fit\_transform(Y)  
     35 KNN\_model = KNeighborsClassifier(n\_neighbors=5, p = 2)  
     36 KNN\_model.fit(x\_train, y\_train)  
     37 print("\*\*\*\*\*K-Nearest Neighbor\*\*\*\*\*")  
     38 print(classification\_report(y\_test, KNN\_model.predict(x\_test), target\_names=le.classes\_))
* Support Vector Machine:
  + 41 def support\_vector\_machine (x\_train, y\_train, x\_test, y\_test):  
     42 Gamma = 0.001  
     43 C = 1  
     44 SVM\_model = SVC(kernel='linear', C=C, gamma=Gamma)  
     45 SVM\_model.fit(x\_train, y\_train)  
     46 print("\*\*\*\*\*Support Vector Machine\*\*\*\*\*")  
     47 print(classification\_report(y\_test, SVM\_model.predict(x\_test)))

**Project Summary:**

I decided to use the three above mentioned classifiers for this project. Not only did I test the data with these three different classifiers, I also experimented with different combinations of features and different proportions of training data to testing data in order to explore different results/possibilities. One key observation I made when originally testing the data was the extremely high accuracy achieved by all three models. All three performed very well with ideal training/testing datasets. Specifically, the test using 70% training data, 30% testing data, and all four features used resulted in a 98% accuracy for all three models. Because this initial test, I decided to test different scenarios because the test was too accurate.

**Scenarios Tested:**

* Scenario 1: 70% training data, 30% testing data, all four features used
* Scenario 2: 40% training data, 60% testing data, all four features used
* Scenario 3: 10% training data, 90% testing data, all four features used
* Scenario 4: 70% training data, 30% testing data, only sepal features used
* Scenario 5: 70% training data, 30% testing data, only petal features used

All of these combinations along with three different classifiers resulted in 15 different tests and results.

**Results Analysis:**

* Scenario 1: 70% training data, 30% testing data, all four features used
  + This testing scenario was interesting while also being problematic. This testing scenario resulted in a 98% accuracy for all three classifiers which is excellent, but because all of the classifiers performed so well, it was hard to decide which classifier performed the best. This test case led me to try different scenarios.
* Scenario 2: 40% training data, 60% testing data, all four features used
  + To test the classifiers further, I decided to reduce the size of the training data to observe its effects on the accuracy of the models. I hypothesized that less training would result in less accurate predictions, but perhaps some of the models would perform better than the others. What I observed was this: The SVM model preformed the best, at a 97% accuracy and little change to its performance before. However, There was a drop in the other two models’ performances, with logistic regression dropping to 94% and KNN dropping to 92%
* Scenario 3: 10% training data, 90% testing data, all four features used
  + To strain the data even more, I reduced the training data to the extremely low percentage of 10%. The resulting accuracies were as follows: SVM at 95%, logistic regression at 91%, and KNN at 90%
* Scenario 4: 70% training data, 30% testing data, only sepal features used and Scenario 5: 70% training data, 30% testing data, only petal features used
  + In these next two scenarios, I decided to test the models with fewer features to see the impact on the accuracy. What I observed was that the accuracy with petals was a tie between all three models at 98%. However, with only sepal features, there was drastically different set of accuracies, which were as follows: logistic regression at 82%, SVM at 80%, and KNN at 73%. What this implies is that the feature most important to the classification of these iris flowers is the petal length and width, with a smaller impact from the sepal length and width.

**Conclusion:**

From the number of different tests (perhaps a bit overkill), I have made a few different conclusions. With varying training size, it would appear that SVM is the most resilient of the three classifiers. I will note that the proportions can’t all be compared directly, since the testing data sizes might have impacted these scores and I probably could have been more thorough and reduced the size of the dataset as a whole, but I would hypothesize a similar effect on the data despite these cases being extremes simply because less training generally results in worse predictions. Sometimes models can get over trained, but I did not observe this phenomenon with this data set probably because the data set size is relatively small.

Another key observation I made was the prediction of Iris Setosa flowers vs the others. The classifiers all did very well classifying these flowers as opposed to the others which they confused occasionally, but I did some research and it would appear that Iris Setosa flowers have the most unique petal and sepal dimensions as opposed to the other two, which have overlapping ranges. This would explain why the models were able to separate these flowers out so well. However, there is enough difference that there was only a few outliers in the data resulting in only a few errors (I’m also not a flower expert, so take my word as fact).

Concerning petal and sepal dimensions, as I mentioned in the analysis, it would appear that petal dimensions are far more important than the sepal dimensions. While it is possible to predict the species based on the sepals, the overall accuracy of all the models was extremely low compared to the tests with petal dimensions. The classifiers all performed well using petal dimensions with accuracies similar to the “ideal” testing case (it’s almost as though the classification would work without sepal dimensions, but further testing would be needed).

My selection for the best performing model is SVM. The SVM model had the highest accuracy or was tied for the highest in four of the five testing scenarios and this demonstrates the versatility of the SVM model. All three models performed well, but I believe that the reason SVM performed the best was because soft margins help the model deal with outliers far better than the other two models even though I only used linear margins in this test. This is a simple data set however, so further testing would reveal SVM’s weakness compared to the other two, but in this specific testing scenario, SVM appears to be the best choice for this simple dataset (SVM is also one of the most popular classifiers). However, with enough data, all of them eventually reached similar results and were able to find the outliers between the Iris Versicolor/Virginica flowers.

**Program Printouts:**

  

