CMPUT466 MINI-PROJECT REPORT

- ## Introduction

  In this project, we are going to complete a task which focuses on a machine learning problem. Based on the dataset of wine, we use a systematic way of hyperparameter tuning to implement a training-validation-test infrastructure. In order to explore the problem better, in addition to comparing the three machine learning algorithms, we will also add a trivial baseline. The three machine learning algorithms including Logistic Regression, SVM and MLP.
  To tune hyperparameters and get the best values, we use validation_curve to do cross-fold validation for three machine learning algorithms. Then we train the model using the best parameters determined in the validation and get their accuracy.
  Comparing the results obtained finally, we can make a conclusion.

- ## Problem formulation

  The dataset we used in this project is “wine.csv”, which is a kind of wine recognition data. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. There are 13 attributes in the data, including Alcohol, Malic acid, etc. All attributes are continuous and there is no missing attribute value. We are trying to predict the three different cultivars labeled as “1, 2, 3” with a total 178 instances. I also printed out the cultivars of wine and their respective quantities in the project. This data is a kind of multiclass classification datasets, which was used with many others for comparing various classifiers. I split data into train and test datasets. And then using scikit-learn to build and train models.
  I got the data from github: https://github.com/jbrownlee/Datasets

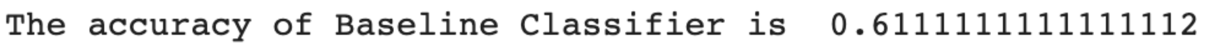
- ## Approaches and baselines

  We compare the three machine learning algorithms and a trivial baseline. The three machine learning algorithms including Logistic Regression, SVM and MLP. Here we train, validate and test these four classifiers, then output the best hyperparameters and accuracy of them respectively.

1. Baseline

We use DummyClassifier with the "most_frequent" strategy from scikit-learn to get a trivial baseline. The "most_frequent" strategy always predicts the most frequent label in the training set.

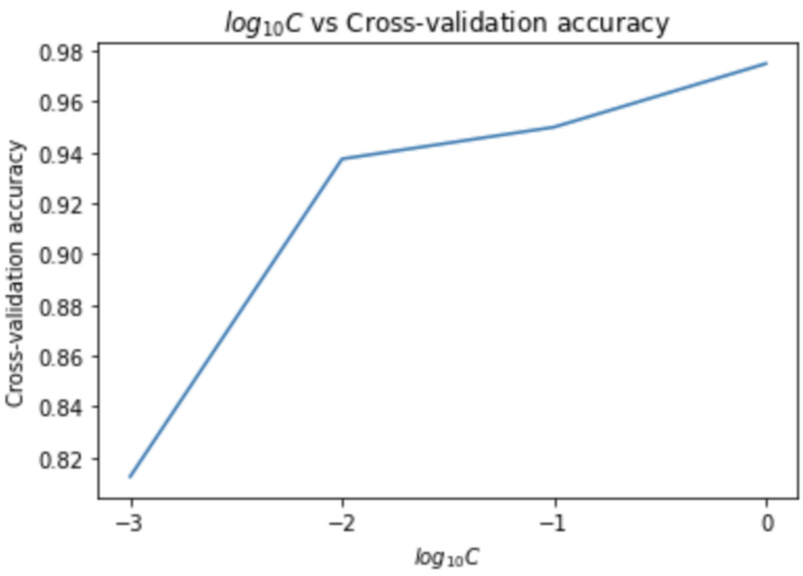
We predict that the accuracy of Baseline performs worse than the other three classifiers.

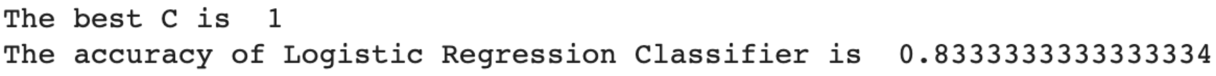


```
The accuracy of Baseline Classifier is  0.6111111111111112
```

2. Logistic Regression

We use LogisticRegression from scikit-learn to get a logistic regression model. At this time we use validation_curve to get cross-fold validation, we can tune the hyperparameter C(Inverse of regularization strength) by validation. After getting the best hyperparameter C, we use it to train the model. By the plot, we can find that the higher the value of C, the better the performance of accuracy.
From the result, we can see that the accuracy of the logistic regression classifier performs better than the baseline classifier.





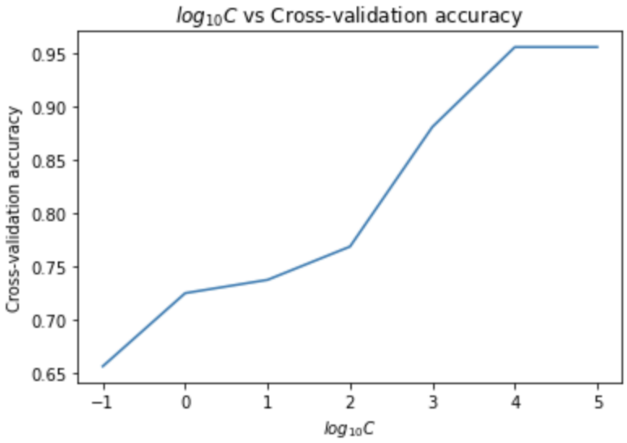
```
The best C is   1
The accuracy of Logistic Regression Classifier is  0.8333333333333334
```

3. SVM

We use SVC from scikit-learn to get a SVM(Support vector machine) model. For this algorithm, using cross_val_score to get cross-validation we can tune the hyperparameter C(Regularization parameter), where C_sets = [0.1, 1, 10, 100, 1000, 10000, 100000]. Control variable and test two different kernels “RBF” and “Linear” respectively.
From the following two plots we can find that the accuracy of the SVM model with two different kernels is close to each other. The accuracy of the SVM classifier with the Linear kernel is a bit higher than the one with the RBF kernel, and it also performs better than the accuracy of the logistic regression classifier. We can speculate that the data we used is relatively linear.

SVM Classifier(RBF kernel):

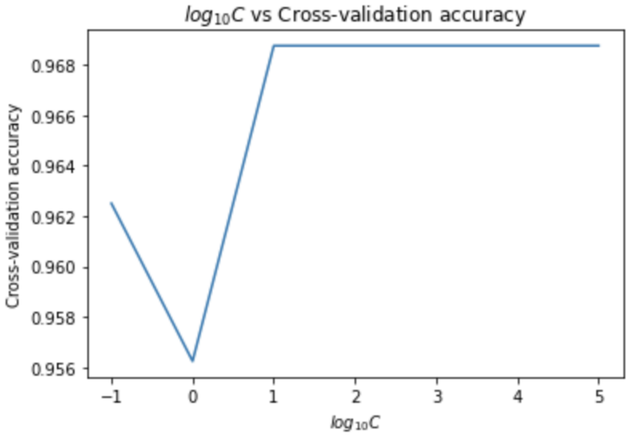
```
The best C is  10000
```



```
The accuracy of SVM Classifier(RBF kernel) is  0.8333333333333334
```

SVM Classifier(Linear kernel):

```
The best C is  10
```

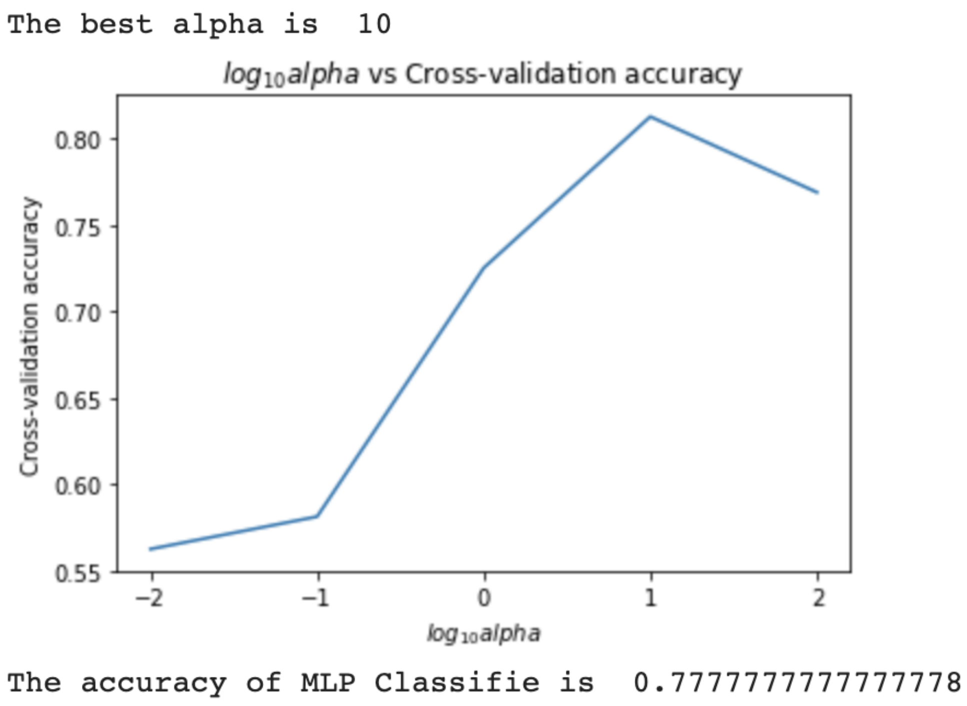


```
The accuracy of SVM Classifier(Linear kernel) is  0.8888888888888888
```

4. MLP

We use MLPClassifier from scikit-learn to get a MLP(Multi-layer Perceptron) model. For this algorithm, using cross_val_score to get cross-validation we can tune the hyperparameter alpha(L2 penalty parameter), where alpha_sets = [0.01, 0.1, 1, 10, 100].

From the result, we can find that the accuracy of the MLP classifier performs a little worse than the logistic regression classifier and SVM classifier.



```
The best alpha is  10
```

$log_{10}alpha$ vs Cross-validation accuracy
Cross-validation accuracy
0.80
0.75
0.70
0.65
0.60
0.55
−2
−1
0
1
2
$log_{10}alpha$

```
The accuracy of MLP Classifie is  0.7777777777777778
```

- **Evaluation Metric**

  We can compare the accuracies of those four different classifiers and find which performs the best. It is a reasonable approximation since we find out the best hyperparameters and use them to train the model.

- **Results**

  According to the results obtained, we can find that the performance of all the models of three machine learning algorithms is significantly better than the baseline classifier. Although the overall performance of the three models is roughly the same, we can see that the accuracy of the SVM classifier with the Linear kernel still performs the best.

  I predict that my dataset is relatively linear because the performance of the logistic regression classifier and the SVM classifier with Linear kernel is particularly similar. Therefore, these two classifiers are good for our dataset to implement a training-validation-test infrastructure.