**Pima Diabetes**

*Question #1a: Run the code once, record the accuracy and AUC score. What do you notice about the scores?*

--ML Model Output--

SVM Acc: 0.76 (+/- 0.04)

SVM AUC: 0.82 (+/- 0.07)

CV Runtime: 0.20750093460083008

This model using SVM classifier achieved 76% accuracy and 0.82 AUC score. CV runtime is the fastest among the all tested classifiers.

*Question #1b: In the Scikit API for SVC, it explains the probability parameter … why did we set it equal to ‘True’? What does that do?*

In classification problem, we want to know what the probability is of assigning target labels. Set 'probability' to true would enable probability estimates.

*Question #2: Run the code once for each setting of the kernel, record the accuracy and AUC scores. What do you notice about the scores compared to Question #1? What about run-times?*

Sigmoid Kernel

--ML Model Output--

SVM Acc: 0.50 (+/- 0.11)

SVM AUC: 0.32 (+/- 0.11)

CV Runtime: 0.2538919448852539

Linear Kernel

--ML Model Output--

SVM Acc: 0.77 (+/- 0.05)

SVM AUC: 0.83 (+/- 0.05)

CV Runtime: 58.34757471084595

Sigmoid kernel has lower accuracy and AUC score compared to rbf kernel while linear kernel has about the same performance as rbf kernel but much longer CV runtime.

*Question #3a: Run the code once, record the accuracy and AUC scores. What do you notice about the scores? How do they compare to the performance Question 2 above for SVMs using a linear kernel with no feature selection?*

--ML Model Output--

SVM Acc: 0.65 (+/- 0.00)

SVM AUC: 0.50 (+/- 0.25)

CV Runtime: 0.09526920318603516

With feature selection, accuracy and AUC score of SVM model using a linear kernel were compromised as both obviously went down, especially AUC with a much higher Standard deviation. However, CV runtime was improved quite a lot. It looks like feature selection is not optimal for this model as it only selected one feature which doesn’t seem sufficient.

*Question #3b: What features were selected, and which were removed? Were there any differences from when you did feature selection with Boosting in HW3, or Random forests in HW2?*

--FEATURE SELECTION ON--

Wrapper Select:

Selected ['Family History']

Features (total/selected): 8 1

Only one feature 'Family History' was selected by SVM model while in random forest model there were four features ('Blood Glucose', 'BMI', 'Family History' and 'Age') and three features in boosting model ('Blood Glucose', 'BMI', 'Age'). It looks like RF model has all the features and what is left by boosting model is kept in SVM model.

**Wine Quality Dataset**

*Question #4a: Run the code once, record the RMSE and Explained Variance.*

--ML Model Output--

SVM RMSE:: 0.77 (+/- 0.06)

SVM Expl Var: 0.08 (+/- 0.10)

CV Runtime: 0.7140622138977051

*Question #4b: In the Scikit API for SVR, you will notice there is no probability parameter (averse to for the classifier version), why do you think that is?*

In regression problem it is not necessary to calculate target variable probability as what we want is the predicted value.

*Question #5: Run the code once for each setting of the kernel, record the RMSE and Explained Variance. What do you notice about the scores compared to Question #4? What about run-times?*

Sigmoid Kernel

--ML Model Output--

SVM RMSE:: 0.91 (+/- 0.22)

SVM Expl Var: 0.00 (+/- 0.00)

CV Runtime: 0.3719189167022705

Linear Kernel

--ML Model Output--

SVM RMSE:: 0.66 (+/- 0.02)

SVM Expl Var: 0.29 (+/- 0.19)

CV Runtime: 15.036526918411255

Sigmoid kernel has higher RMSE and lower explained variance compared to rbf kernel while linear kernel has better performance than rbf kernel but much longer CV runtime.

*Question #6: Run the code once, record the RMSE and Explained Variance. What do you notice about the scores, or the run times? How do they compare to results in Question #5?*

--ML Model Output--

SVM RMSE:: 0.66 (+/- 0.02)

SVM Expl Var: 0.29 (+/- 0.16)

CV Runtime: 0.7831878662109375

After normalization, model performance didn’t change much but CV runtime was improved from 15s to 0.78s. It is necessary to normalize variables before modeling with SVM classifier.

*Question #7a: Run the code once Record the RMSE and Explained Variance Score. What do you notice about the scores? How do they compare to performance in Question 6 above for SVMs using a linear kernel with no feature selection?*

--ML Model Output--

SVM RMSE:: 0.68 (+/- 0.04)

SVM Expl Var: 0.27 (+/- 0.25)

CV Runtime: 0.39678168296813965

Performance similar to model with all features in Question #6. There was a slight drop in performance, but not outside the expected range of variation so the performance of the models cannot be considered different. CV runtime was reduced to using half time of all features. In other words we were able to build a model with less features using SVM classifier that had similar performance.

*Question #7b: What features were selected, and which were removed?* *How did this compare with features selected in previous homeworks (Random Forests, Gradient Boosting)?*

--FEATURE SELECTION ON--

Wrapper Select:

Selected ['volatile acidity', 'sulphates', 'alcohol']

Features (total/selected): 11 3

The selected three features ('volatile acidity', 'sulphates', 'alcohol') are same as previous classifiers. In other words these three variables are critical in determining wine quality.

*Question #8a: Run the code once Record the RMSE and Explained Variance Score. What do you notice about the scores? How do they compare to performance above using wrapper feature selection in Question 7?*

--ML Model Output--

SVM RMSE:: 0.67 (+/- 0.03)

SVM Expl Var: 0.27 (+/- 0.22)

CV Runtime: 0.47161293029785156

Performance similar to model with wrapper feature selection in Question #7 so the performance of the models cannot be considered different. CV runtime was slightly more compared with wrapper method.

*Question #8b: What features were selected, and which were removed?*

--FEATURE SELECTION ON--

Univariate Feature Selection - Mutual Info:

Ranked Features

0 alcohol : 0.1635624634675512

1 volatile acidity : 0.1280698726796885

2 sulphates : 0.1075129499572336

3 density : 0.09766884325929759

4 total sulfur dioxide : 0.08472413399741097

5 citric acid : 0.07683307120847793

6 fixed acidity : 0.06079843114247119

7 free sulfur dioxide : 0.03558387789083195

8 chlorides : 0.030080280567937656

9 residual sugar : 0.022210548328622792

10 pH : 0.0

Selected ['volatile acidity', 'total sulfur dioxide', 'density', 'sulphates', 'alcohol']

Features (total/selected): 11 5

Besides three common features in wrapper method, two more features were selected by univariate selection method, which are 'total sulfur dioxide' and 'density'.

*Question #9a: Run the code once Record the RMSE and Explained Variance Score. What do you notice about the scores? How do they compare to performance above for feature selection using the simple wrapper in Question 7 and the univariate mutual info in Question 8?*

--ML Model Output--

SVM RMSE:: 0.66 (+/- 0.02)

SVM Expl Var: 0.30 (+/- 0.18)

CV Runtime: 0.6869709491729736

Performance similar to model with wrapper feature selection in Question #7 and univariate selection in Question #8 so the performance of the models cannot be considered different. CV runtime was the longest among three methods. In other words wrapper is the most efficient feature selection method on thisdataset.

*Question #9b: What features were selected, and which were removed?*

# of Feature Combos Tested: 231

0.6596825116948701 [0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1] 11

Wrapper Feat Sel Runtime: 93.78639793395996

Selected ['volatile acidity', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol']

Features (total/selected): 11 9

9 features were selected, and they are 'volatile acidity', 'residual sugar', 'chlorides', 'free sulfur dioxide', 'total sulfur dioxide', 'density', 'pH', 'sulphates', 'alcohol'. Only 2 features were removed which are 'fixed acidity' and 'citric acid'.

*Question #10a:* *Based on the comment on line 56, explain how we might search the feature set space in a more optimal manner.*

Currently search method is using brute-force search which test each combination. Use greedy search or random search would improve searching runtime.

*Question #10b: If you uncomment the print statements on lines 87 and 96, and watch the code run, you may notice that there are actually several different feature sets that perform nearly the same as the optimal feature set, some of which have much fewer features than others. What is one way we could force the wrapper method to select smaller feature sets, even if they have slightly less performance? (HINT: line 56 also mentions something about this)*

Use regularization L1 or L2 to add penalties and early stopping to terminate bad candidate feature subsets and avoid wasting of training time.

**Summary Questions**

*Question #11: Line up the results from Homeworks 1,2,3,4 as a table for both Diabetes and Wine, with rows of performance metrics for each ML method (Decision Trees, Random Forests, Gradient Boosting, Ada Boost, Neural Networks, SVMs). Looking at this table of performance metrics, how would you explain the table to a boss or customer.*

*Diabetes Dataset*

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier Name | Accuracy | AUC | CV Runtime |
| Decision Tree | 0.71 (+/- 0.08) | 0.69 (+/- 0.07) | 0.0271 |
| Random Forest | 0.77 (+/- 0.08) | 0.83 (+/- 0.07) | 0.5776 |
| Random Forest - FS | 0.76 (+/- 0.05) | 0.82 (+/- 0.06) | 0.6007 |
| Gradient Boosting | 0.76 (+/- 0.07) | 0.82 (+/- 0.06) | 0.2859 |
| Gradient Boosting – FS | 0.77 (+/- 0.05) | 0.83 (+/- 0.07) | 0.2374 |
| Ada Boost | 0.76 (+/- 0.05) | 0.83 (+/- 0.06) | 0.6124 |
| Neural Network – lbfgs | 0.70 (+/- 0.05) | 0.71 (+/- 0.05) | 1.3053 |
| Neural Network – Adam | 0.72 (+/- 0.07) | 0.80 (+/- 0.08) | 2.1903 |
| SVM – rbf kernel | 0.76 (+/- 0.04) | 0.82 (+/- 0.07) | 0.2075 |
| SVM – sigmoid kernel | 0.50 (+/- 0.11) | 0.32 (+/- 0.11) | 0.2539 |
| SVM – linear kernel | 0.77 (+/- 0.05) | 0.83 (+/- 0.05) | 58.3476 |
| SVM – linear kernel FS | 0.65 (+/- 0.00) | 0.50 (+/- 0.25) | 0.0953 |

*Wine dataset*

|  |  |  |  |
| --- | --- | --- | --- |
| Regressor Name | RMSE | Explained Variance | CV Runtime |
| Decision Tree | 0.90 (+/- 0.10) | -0.31 (+/- 0.17) | 0.0461 |
| Random Forest | 0.65 (+/- 0.02) | 0.33 (+/- 0.11) | 0.8945 |
| Random Forest - FS | 0.68 (+/- 0.03) | 0.25 (+/- 0.11) | 0.5873 |
| Gradient Boosting | 0.64 (+/- 0.01) | 0.34 (+/- 0.12) | 0.3039 |
| Ada Boost | 0.66 (+/- 0.03) | 0.31 (+/- 0.16) | 0.9848 |
| Neural Network | 0.66 (+/- 0.05) | 0.29 (+/- 0.17) | 1.8055 |
| SVM – rbf kernel | 0.77 (+/- 0.06) | 0.08 (+/- 0.10) | 0.7141 |
| SVM – sigmoid kernel | 0.91 (+/- 0.22) | 0.00 (+/- 0.00) | 0.3719 |
| SVM – linear kernel | 0.66 (+/- 0.02) | 0.29 (+/- 0.19) | 15.0365 |
| SVM – linear kernel (norm feature) | 0.66 (+/- 0.02) | 0.29 (+/- 0.16) | 0.7832 |
| SVM – linear kernel (wrapper selection) | 0.68 (+/- 0.04) | 0.27 (+/- 0.25) | 0.3968 |
| SVM – linear kernel (univariate selection) | 0.67 (+/- 0.03) | 0.27 (+/- 0.22) | 0.4716 |
| SVM – linear kernel (exhaustive selection) | 0.66 (+/- 0.02) | 0.30 (+/- 0.18) | 0.6870 |

For both datasets, Random Forest and Gradient Boosting methods would achieve similar performance with slightly different CV runtime.

Decision tree could achieve similar performance as Random Forest and Gradient Boosting on classification with better in CV runtime with slightly lower accuracy. However, it does poorly on regression problem like wine dataset.

*Which algorithm is best depends on the needs. If the goal is to achieve highest accuracy regardless of computation time, we could try either* Random Forest or Gradient Boosting. If the goal is to get a reliable and interpretable result within minimum time, then we could go with decision tree.

*Question #12: If we had to explain to someone what really drives peoples’ perception of wine quality, what would you say based on your findings in this homework (e.g. Q8) and previous ones? Are there 2-3 features we can say are consistently most important? If so, can you hypothesize why those features might be important?*

Volatile acidity, sulphates, alcohol are consistently important features in determining wine quality confirmed by various algorithms.

* Volatile acidity refers to the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste. This is caused by a type of bacterial spoilage which produces large amounts Acetic acid (vinegar) which is a serious wine fault.
* Sulphates is a wine additive which can contribute to sulfur dioxide gas (S02) levels, which acts as an antimicrobial and antioxidant.
* Alcohol refers to the percent alcohol content of the wine, which can produce ethyl acetate which smells like nail-polish remover and is also a serious wine fault.

Wines that have good balance will be of higher quality than ones where one component stands out above the rest. The five components – acidity, tannins, sugar/sweetness, alcohol and fruit – need to be balanced. Higher sulfurous content causes a duller taste in wine, and that high potency of sulfite ions presents a health risk and speeds up the wine’s fermentation process. This suggests that higher sulphate contents tend to correspond with lower wine quality.