exercise1

March 23, 2024

This exercise was performed on ARCC Beartooth[?] using the following environment/software packages: Conda Env Exported to: lreilly-bt-ml.yml

Load Preinstalled Modules: module load gcc12.2.0 miniconda3 git/2.37.0

Create environment then activate with: conda env create -f whatev_ml_env.yml
conda activate whatev_ml_env

Make sure jupyter can see the kernel in your conda env so you can select from southpass interface dropdown: python -m ipykernel install –user –name=whatev_conda_torch

Download data to folder and extract wget https://archive.ics.uci.edu/static/public/186/wine+quality.zip
unzip wine+quality.zip

```
Python Platform:macOS-14.0-arm64-arm-64bit
Python 3.10.12 | packaged by conda-forge | (main, Jun 23 2023, 22:41:52) [Clang
15.0.7 ]
PyTorch Version:  2.1.1
Pandas Version:  2.1.4
SkLearn Version:  1.3.0
```

1 Warmup

Download the Wine Quality dataset. Choose the one that corresponds to your preference in wine.

Downloaded and unzipped to folder/repo

[4]:
```
#Ensure we're looking at the right place
print(os.getcwd())

#Import data and separate out
data = pd.read_csv('winequality-red.csv', sep=';')

#view data, get info about it, clean if necessary
data.info()
data.describe().T
```

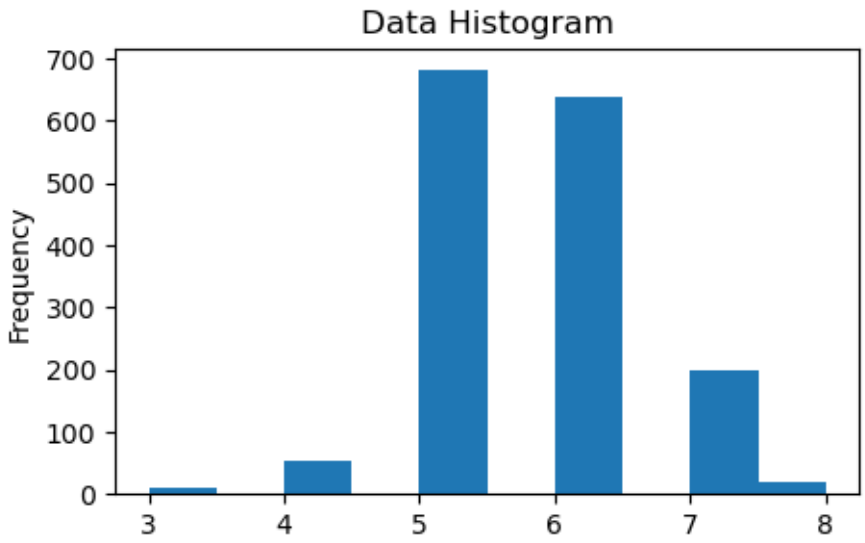
[4]:

	count	mean	std	min	25%	\
fixed acidity	1599.0	8.319637	1.741096	4.60000	7.1000	
volatile acidity	1599.0	0.527821	0.179060	0.12000	0.3900	
citric acid	1599.0	0.270976	0.194801	0.00000	0.0900	
residual sugar	1599.0	2.538806	1.409928	0.90000	1.9000	
chlorides	1599.0	0.087467	0.047065	0.01200	0.0700	
free sulfur dioxide	1599.0	15.874922	10.460157	1.00000	7.0000	
total sulfur dioxide	1599.0	46.467792	32.895324	6.00000	22.0000	
density	1599.0	0.996747	0.001887	0.99007	0.9956	
pH	1599.0	3.311113	0.154386	2.74000	3.2100	
sulphates	1599.0	0.658149	0.169507	0.33000	0.5500	
alcohol	1599.0	10.422983	1.065668	8.40000	9.5000	
quality	1599.0	5.636023	0.807569	3.00000	5.0000	

	50%	75%	max
fixed acidity	7.90000	9.200000	15.90000
volatile acidity	0.52000	0.640000	1.58000
citric acid	0.26000	0.420000	1.00000
residual sugar	2.20000	2.600000	15.50000
chlorides	0.07900	0.090000	0.61100
free sulfur dioxide	14.00000	21.000000	72.00000
total sulfur dioxide	38.00000	62.000000	289.00000
density	0.99675	0.997835	1.00369
pH	3.31000	3.400000	4.01000
sulphates	0.62000	0.730000	2.00000
alcohol	10.20000	11.100000	14.90000
quality	6.00000	6.000000	8.00000

[5]:
```
print(data.head())
```

[6]:
```
#data looks good, export and pull label data from rest of the dataset
labels = data.pop("quality")
display(labels) #should be a list of int64s

#Show a histogram of the distribution of label data
plot = labels.plot(kind='hist', title="Data Histogram", figsize=(5,3))
```



1.1 Regression

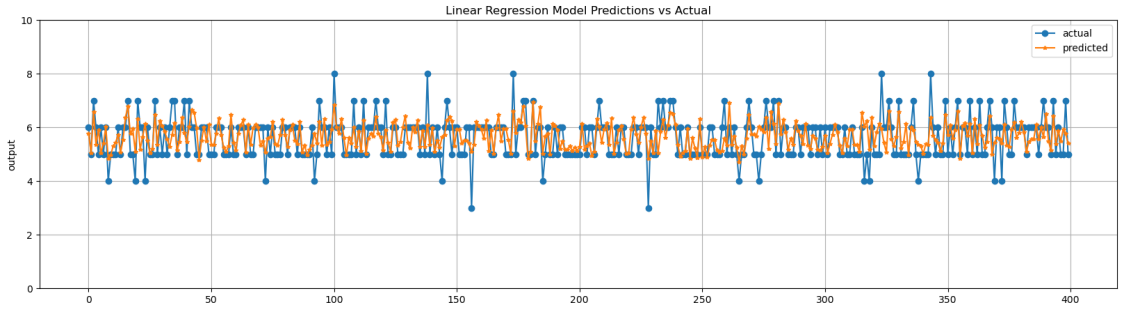
Build a regression model to predict the wine quality. You can choose any model type you like; the purpose of this exercise is to get you started. Evaluate the performance of your trained model – make sure to get an unbiased performance estimate!

[7]:
```
#Initial Regression Model to Predict Wine Quality Using sklearn linear␣
 ↪regression prepackaged ML
#https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.
 ↪LinearRegression.html

#AKA'ing our data into X features and y labels
#Uses train_test_split for model evals and use to shuffle our data and randomize␣
 ↪what gets parsed to test and train
X_train_raw, X_test_raw, y_train, y_test = train_test_split(data, labels,␣
 ↪train_size=.75, shuffle=True,random_state=0)

X_train_raw.shape, X_test_raw.shape
```

[7]:
```
((1199, 11), (400, 11))
```

```
The Linear Regression Model Training Accuracy Score on is:  0.3632493675603261
The Linear Regression Model Explained Variance Score is:  0.3500683511219984
The Mean Squared Error for this Linear Regression Model is:  0.40007252708505525
the R2 Score for this Linear Regression Model is:  0.3454243812456276
```

```
<Figure size 640x480 with 0 Axes>
```




```
The Linear Regression Model Explained Variance Score is:  0.25147508461923884
The Mean Sqared Error for this Linear Regression Model is:  0.4575
the R2 Score for this Linear Regression Model is:  0.251464858729331
```

1.2 Submission

Upload your code and a brief description of your results.

1.2.1 Linear Regression Model Evaluation:

This initial exercise uses the most basic of models (A Linear Regression Model). It is recognized that this is far from an ideal model for this particular prediction problem and dataset. The idea was to show a very naively developed model, which should result in poor performance, and then see how to improve upon it.

Metrics Used: The Linear Regression Model Training Accuracy Score is: 0.5963302752293578 The Accuracy of the Test Data Predictions is: 0.6275 the R2 Score for this Logistic Regression Model is: 0.19829023120737088 For the Linear Regression Model, we use the following metrics for model evaluation. 1. Model Prediction Score: Per SciKitLearn's Website, The accuracy() method "returns the mean accuracy on the given data and labels." Additionally, when performing multi label classification, "this references the subset accuracy which is a harsh metric since it requires each that each label set be correctly predicted for each sample". This particular score is used in this situation to determine the accuracy of prediction on the training data.
2. Explained Variance Score: Per SciKitLearn's Website, This metric essentially represents the amount of variation in the original dataset that our model is able to explain. 3. MSE Score: Per SciKitLearn's Website, this calculates the mean squared error regression loss on the model.

4. R2 Score: Per SciKitLearn's Website, this returns the coefficient of determination on the regression model.

Accuracy: The above scores are all metrics to compute the accuracy of the overall model. Run output is as follows:

Additionally, the linear regression model in general is very bad at predicting lower quality wines. All predicted values tend to fall between 5 and 7, and no values under quality 5 are ever predicted correctly.

Summary: The simple Linear Regression model is a classical model providing a line for ordinary least squares linear regression on the dataset. The model as developed is not a good predictor of wine quality and fails to accurately predict true wine quality in a majority of cases on test data.

1.2.2 Logistic Regression Model Evaluation

This regression exercise uses a logistic regression model detailed [here] (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html). This does a much better job of predicting values than the prior Linear Model. Additionally, the logistic regression model uses the [model_selection.GridSearchCV() Optimizer] (https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridS "to implement a fit and score method and apply them optimized by cross validated grid-search over the parameter grid" that is specified. Ideally, this should come up with the best set of hyperparameters from those listed in the parameter grid to optimize the model for Logistic Regression. The specific hyperparameters are selected because they support classification of multi-class datasets.

Metrics Used: For the Logistic Regression Model, we use the following metrics for model evaluation. 1. Model Prediction Score: Per SciKitLearn's Website, The accuracy() method "returns the mean accuracy on the given data and labels." Additionally, when performing multi label classification, "this references the subset accuracy which is a harsh metric since it requires each that each label set be correctly predicted for each sample".

2. Accuracy Score: Per SciKitLearn's Website, This metric represents the mean accuracy on the given test data and labels given the model as set.
3. MSE Score: Per SciKitLearn's Website, this calculates the mean squared error regression loss on the model.
4. R2 Score: Per SciKitLearn's Website, this returns the coefficient of determination on the regression model.

Accuracy: The above scores are all metrics to compute the accuracy of the overall model. Run output is as follows:

1. The Logistic Regression Model Training Accuracy Score is: 0.5963302752293578
2. The Accuracy of the Test Data Predictions is: 0.6275
3. The R2 Score for this Logistic Regression Model is: 0.19829023120737088
4. The Mean Squared Error for this model is: 0.49

Evaluation metrics on this model are improved from the linear regression model. It however, like the linear regression model does not do a great job of predicting outliers. All predicted values tend to fall between 5 and 7.

Summary: The Logistic Regression model performs a logistic regression using a list of solvers then selected from optimized parameters within the parameter grid. The model as developed is still not a good predictor of wine quality and fails to accurately predict true wine quality in a majority of cases on test data.

1.2.3 Classification Model Evaluation

This particular classification model was designed using a classification support vector machine (svc-SVM). Two different kernel selections were made but other hyperparameters remain constant. Our kernel = linear in our first run, and kernel = rbf (radial) in the second. In both classification models, we set gamma = auto, and C = 5.

In the first case of classification a linear kernel is selected as it is the most simple of kernel functions using a linear decision plane.

The second case of classification we selected a radial kernel (rbf). This kernel requires C and gamma. C is our trade off value for misclassifying. Giving C a lower value will make our decision surface "softer" while a higher value for C sets a higher value for classifying our training examples correctly. Setting the value to 5 is somewhat on the higher side, but not so rigid as to overfit our model to the training data.

Metrics Used:

1. Classification Score: Per SciKitLearn's Website, The score() method "returns the mean accuracy of the model prediction on the given test data and labels". Furthermore, "In multilabel classification, this is the subset accuracy which is a harsh metric since it requires each label set be correctly predicted" (similar to the linear regression score function).
2. Accuracy Score: Per SciKitLearn's Website, This score metric "returns the fraction of correctly classified samples".
3. Confusion Matrix: Per SciKitLearn's Website, The confusion matrix is used to evaluate the accuracy of our classification. The matrix shows true labels on on axis and predicted labels on the other axis, and provides a count for the number of predictions of each type vs what their true label type is.

Accuracy: Since we ran two separate SVC models with different kernels, we can analyze the accuracy of each,

Linear Kernel

Beginning with the Linear Kernal, we can see that the model performs alright. Certainly much better than our linear regression model. Evaluation metric outputs are as follows:

1. The classification score for this Classification Model is: 0.5796497080900751
2. The accuracy for this Classification Model is: 0.615

On our first look, we can certainly see performance has improved with the support vector machine compared to linear regression. Performance, however is still not great. Under most circumstances, a "good" machine learning model should perform with accuracy around 80% or above. Furthermore, our test dataset consists of a total of 400 wines. Of these wines, the linear kernel classifies 131 wines of quality = 4 correctly, and 115 wines of quality = 5 correctly. It however fails to classify any wines correctly if they fall outside those labels and only predicts their quality as either 4 or 5. 61 wines in the dataset have wine quality falling outside the common quality labels 4 and 5 and all of these wines are categorized incorrectly. Therefore the model performs badly on all outliers.

Radial Kernel

In our second iteration of the SVC model, we use a radial kernel. This model results with the following metrics:

1. The classification score for this Classification Model is: 0.8940783986655546
2. The accuracy for this Classification Model is: 0.55

Our classification score is looks very high. Our total accuracy however is lower than that of the linear kernel. The confusion matrix shows that 115 wines of quality 4 are predicted accurately, and 87 wines of quality 5 are predicted accurately. In addition, the radial model predicts 18 wines of quality 6 correctly. The radial model does appear to perform better on outliers and does try to predcit outliers whereas the linear model would not.

Summary: The radial kernel performs well better on outlying data but we compromise overall accuracy. The linear model performs with better accuracy but fails on all outliers. In general the SVC performs far better than a simple Linear Regression classifier.

References

[1] Advanced Research Computing Center (2023) Beartooth Computing Environment, x86_64 cluster. University of Wyoming, Laramie, WY https://doi.org/10.15786/M2FY47

[2] A. Asuncion, D. Newman, UCI Machine Learning Repository, University of California, Irvine (2007). Obtained from https://archive-beta.ics.uci.edu/dataset/186/wine+quality.

[3] C. Harris, K. Millman, S. van der Walt, Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2. https://numpy.org/doc/stable/reference/generated/numpy.isnan.html

[4] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.