ARAC Lab Interview Problem Task

Edward Sung
4/17/22

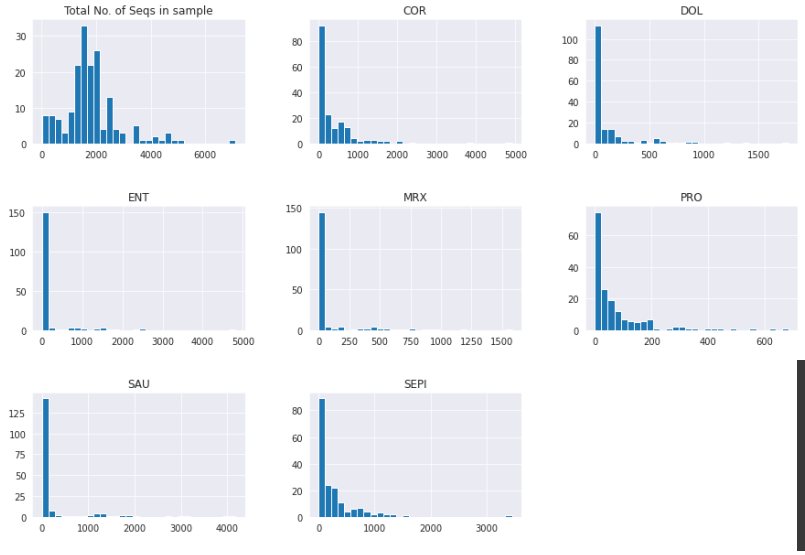
Raw Data Set Evaluation

There are 178 samples with 16 feature columns. “Shareable Sample Name” column is a reference name that can be dropped. CST1 – CST7 are cluster IDs that can be combined into a single column to be used as the target label for classifying. This leaves 8 columns, that refer to select prokaryotes at various taxonomy levels, to be used as features for the modeling. There are no missing data.

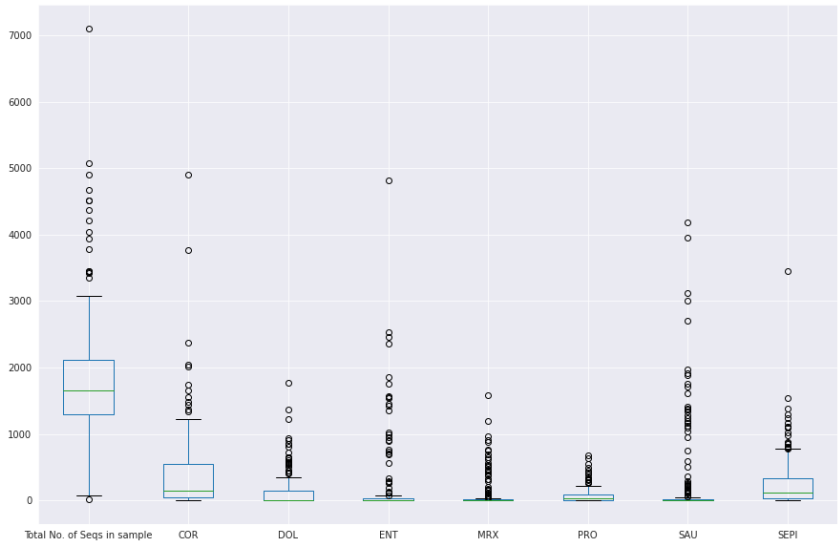
Note any graphics displayed here can be found in my notebook in my github.

EDA on Whole Data

Each of the feature columns, except for No. of Seq, which isn’t as heavily skewed, follow a heavy right skewed distribution as shown in the histogram distribution and positive skewness calculations. This can also be seen in the boxplot with a significant number of outliers marked in the upper range. X-axis is feature value, y-axis is sample count for the histogram.



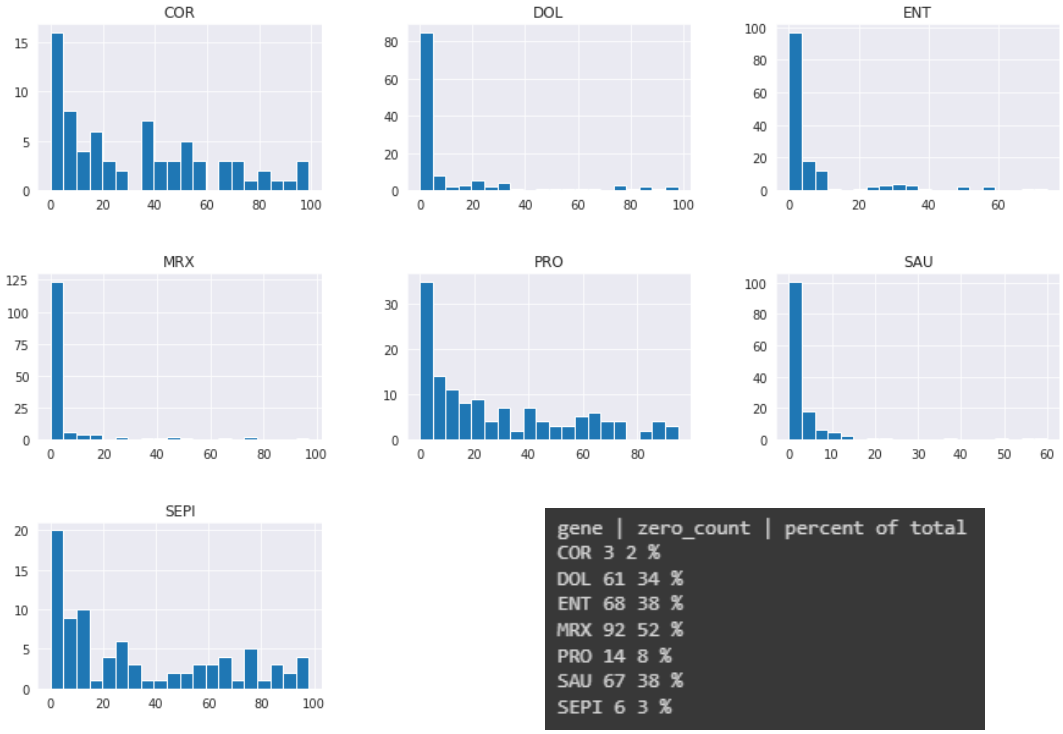

```
feature | Skewness
Total No. of Seqs in sample 1.4017596404493577
COR 3.7887130974099157
DOL 2.9657125351074574
ENT 4.3215090691711335
MRX 3.3248176525705904
PRO 2.668058502939298
SAU 3.337262903372747
SEPI 3.64961596032243
```



ARAC Lab Interview Problem Task

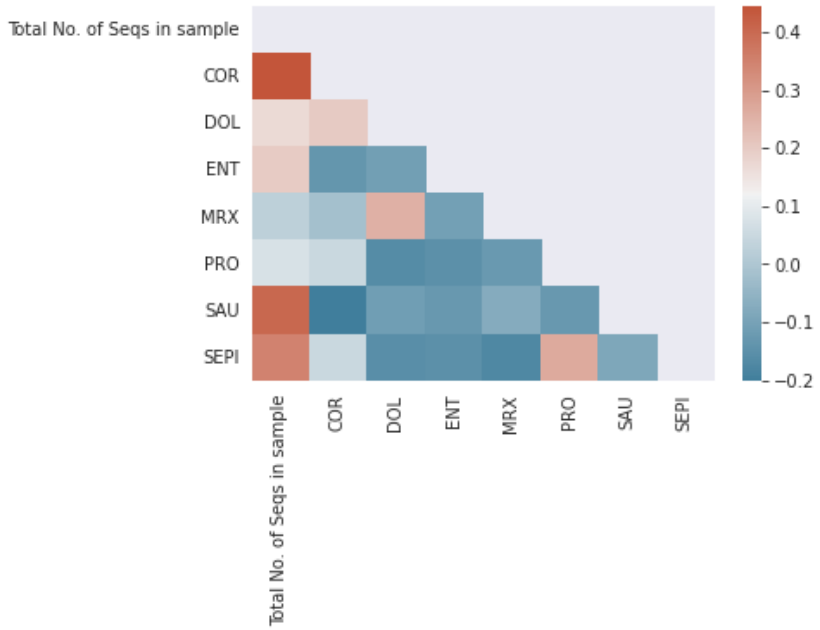
Edward Sung
4/17/22

Upon closer evaluation and counting of values on the lower end, there are a significant number of samples valued at or close to 0 for their feature, such as “MRX” having 52% of its values be 0. X-axis is feature value under 100, y-axis is sample count.



gene	zero_count	percent of total
COR	3	2 %
DOL	61	34 %
ENT	68	38 %
MRX	92	52 %
PRO	14	8 %
SAU	67	38 %
SEPI	6	3 %

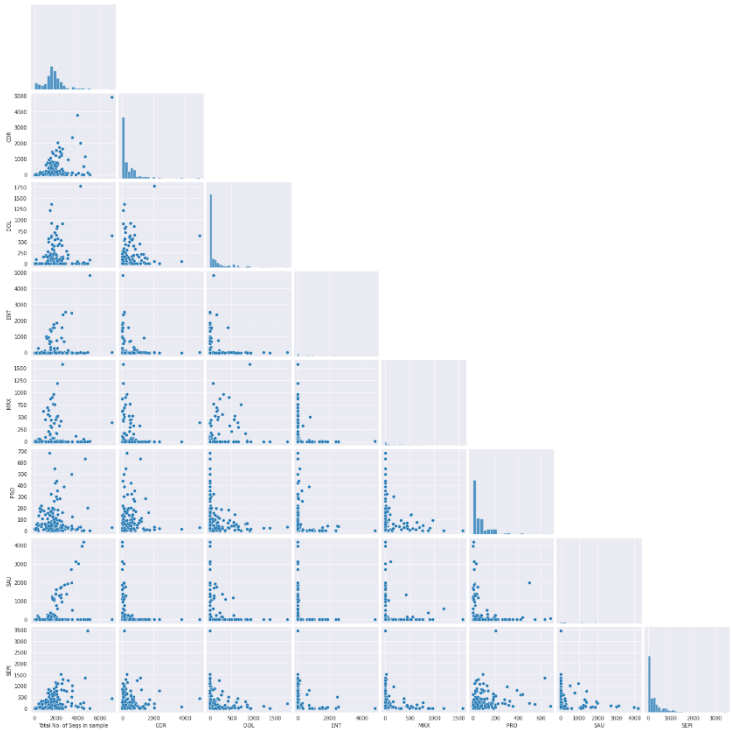
A heatmap on the Pearson’s correlation shows that there is a significant positive correlation between COR and SAU with No. of Seqs and a significant negative correlation between SAU and COR, SEPI and MRX.



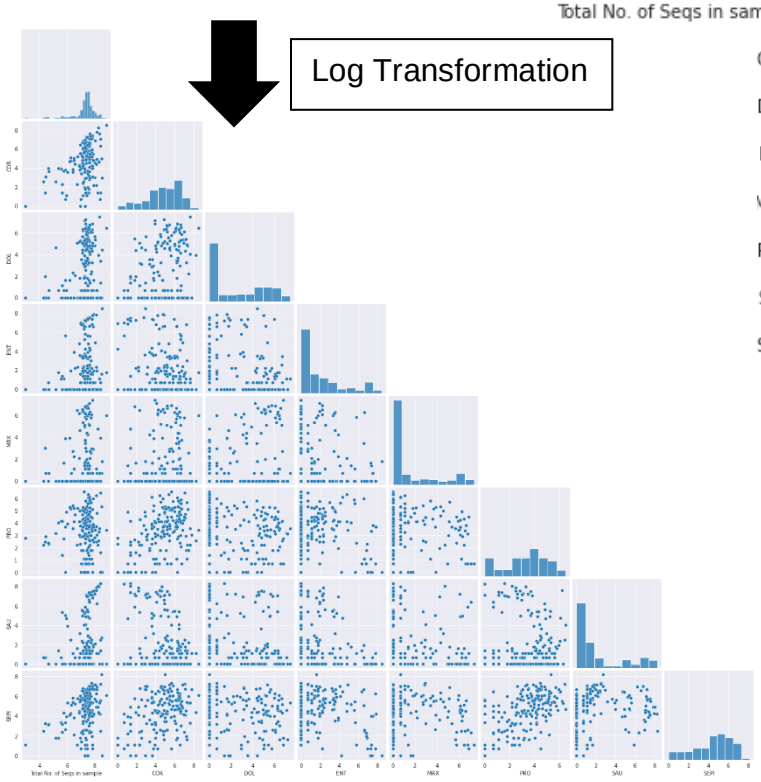
ARAC Lab Interview Problem Task

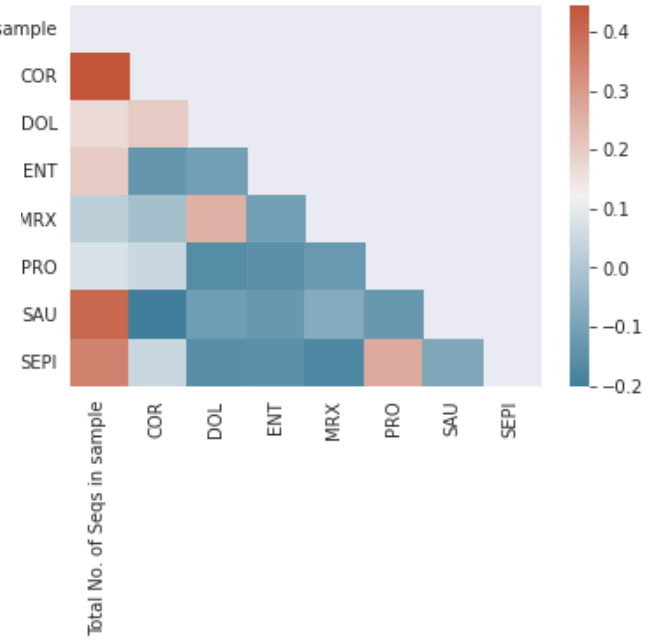
Edward Sung
4/17/22

But due to the heavy skewness, it is difficult to observe this in the original data pairplot. But since they all follow a positive skew distribution, a log transformation can separate the points and declutter the plots. In the log transformed dataset, it is easier to see the relation between the pairplot and heatmap of the Pearson's correlation values. Heatmap copied down for comparison.



Log Transformation

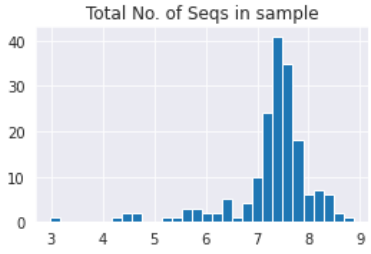


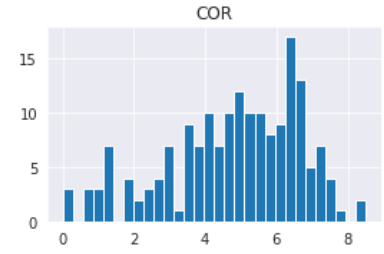


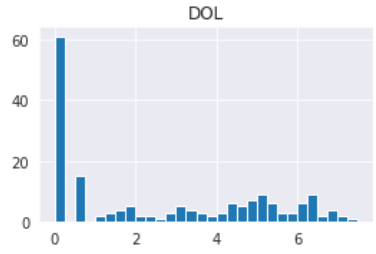
ARAC Lab Interview Problem Task

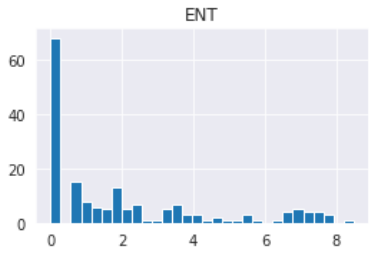
Edward Sung
4/17/22

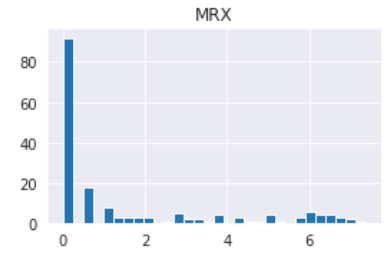
Graphs for the log transformed data. It shows a better visualization and spread of the distribution of data.

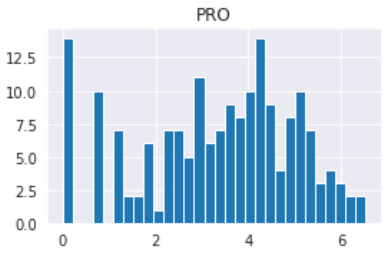


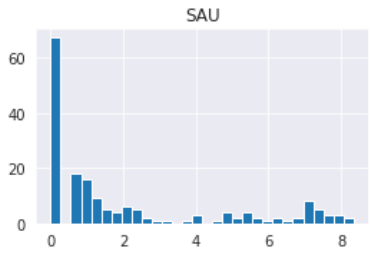


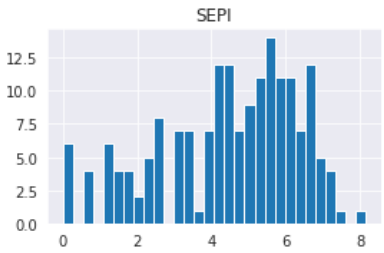




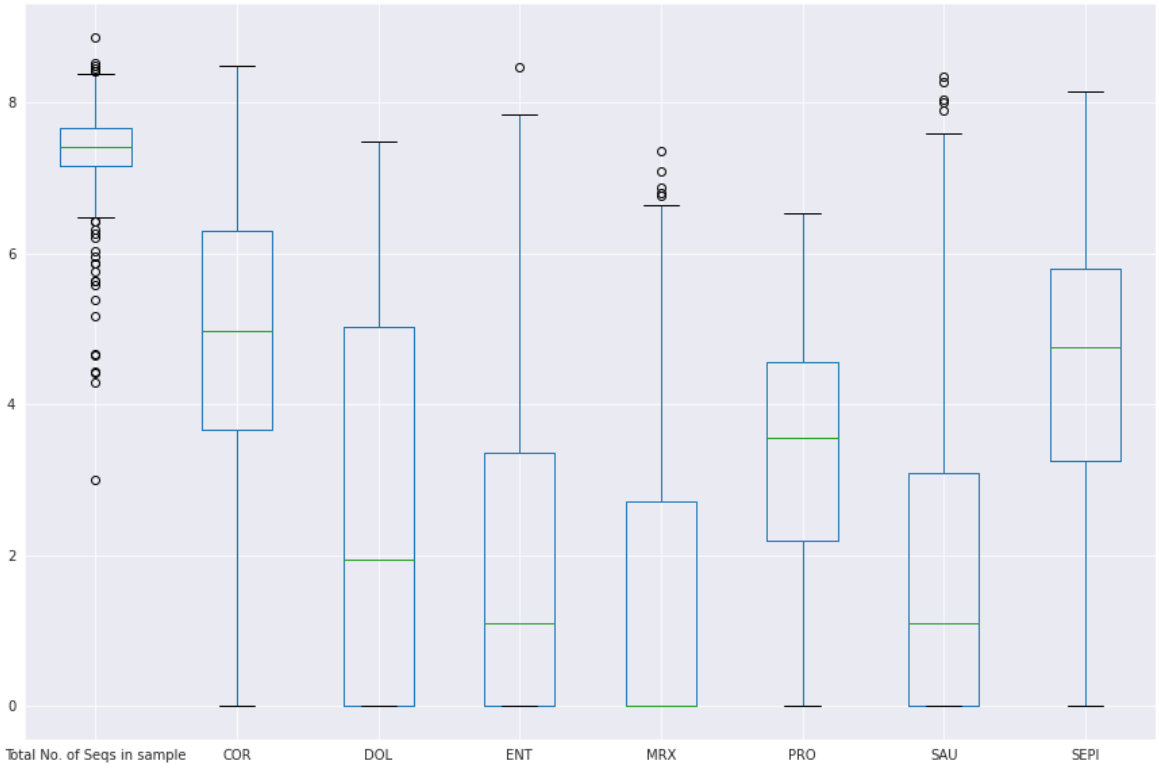








feature	Skewness
Total No. of Seqs in sample	-2.065245249423395
COR	-0.5733218744064065
DOL	0.31399018984082766
ENT	1.1155079379616037
MRX	1.2204604309014933
PRO	-0.38913924319214677
SAU	1.1183575359335856
SEPI	-0.619676220504361 3

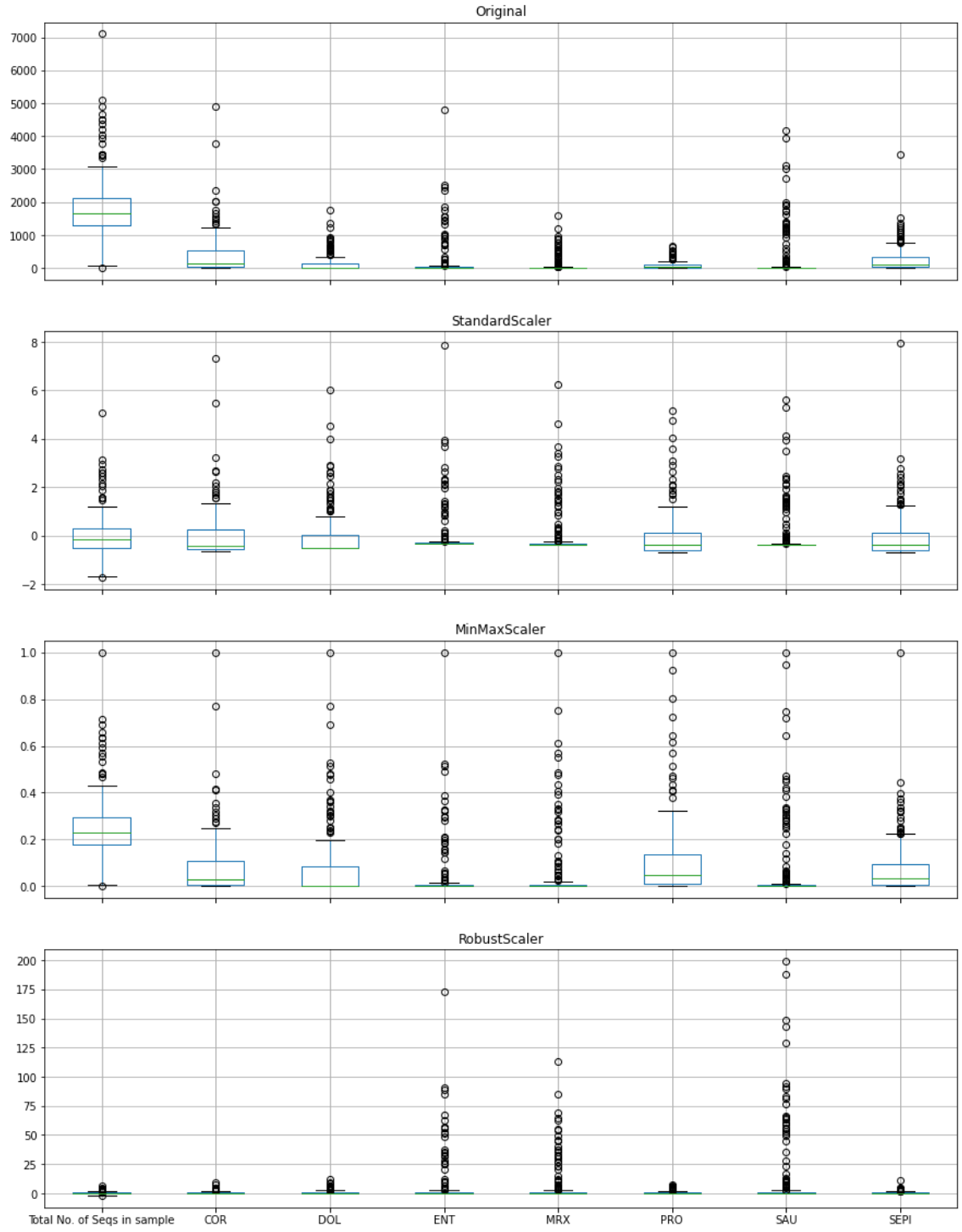


ARAC Lab Interview Problem Task

Edward Sung
4/17/22

Normalizing Data to Consider

There are a few popular ways to normalize data for features to be comparable with one another as well as have similar impact on the modeling that is provided by scikit-learn. There is StandardScaler, MinMaxScaler, and RobustScaler. These different feature scalers can be evaluated against each other and picked for the best one on the dataset. There are also power transformations like box-cox and yeo-johnson that can make the data set more Gaussian-like.

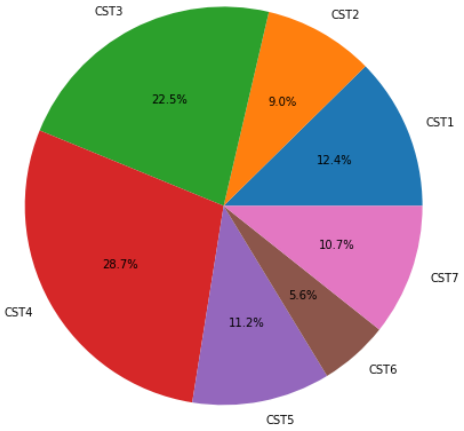


ARAC Lab Interview Problem Task

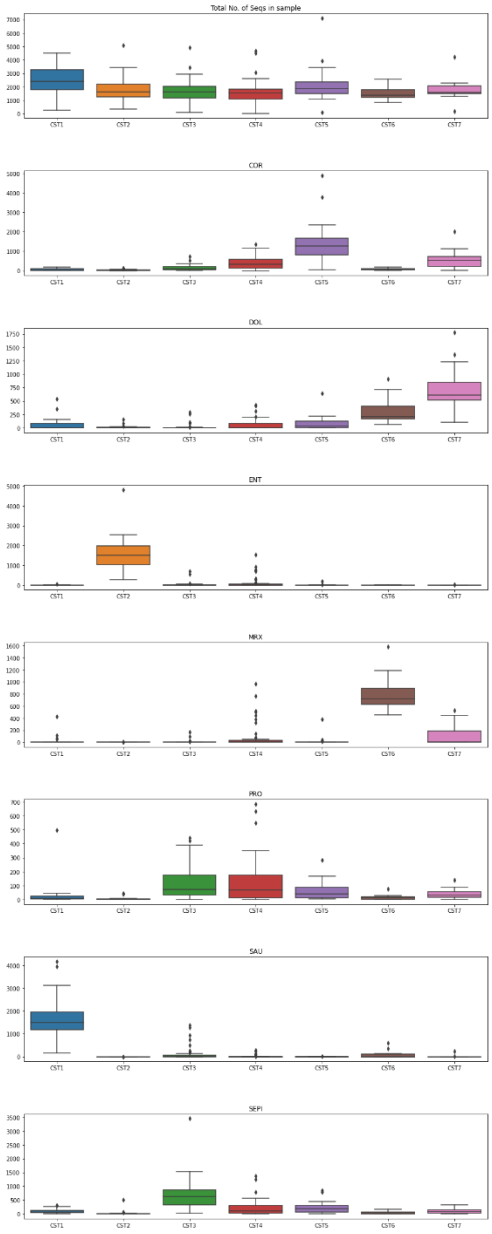
Edward Sung
4/17/22

EDA on Cluster Data

There is a class imbalance between the clusters to consider for the modeling, with CST4 being about 5 times more than CST6.



cstID	Count
CST1	22
CST2	16
CST3	40
CST4	51
CST5	20
CST6	10
CST7	19



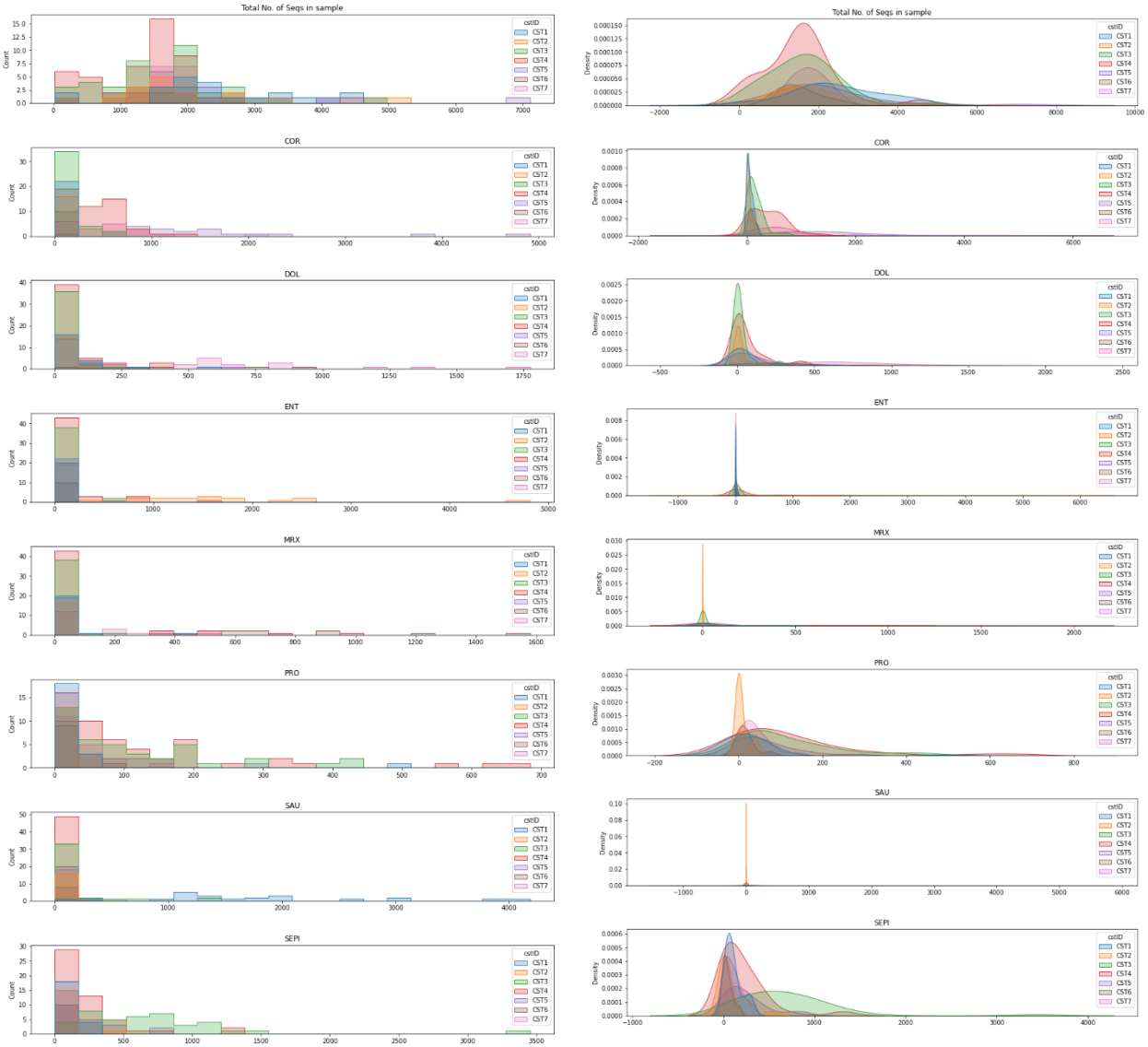
Following-up on the noted skewed distribution in the EDA on the whole data, when the data is graphed by clusters, it shows that the majority cases the bacteria groups are on the low end for values, except for specific bacteria groups to a clusterID that accounts for the higher end outliers. In other words, as seen in the boxplots, most average means are towards the low end, except for one or two clusters that hold a high value amount for the bacteria group.
Examples to note:

- CST1 has the highest average sequence length.
- CST5 has the highest average COR values.
- CST7 has the highest average DOL values.
- CST2 has the highest average ENT values.
- CST6 has the highest average MRX values.
- CST3 and CST4 have similar average PRO values.
- CST1 has the highest average SAU values, but also a large range of values.
- CST3 has the highest average SEPI values.

ARAC Lab Interview Problem Task

Edward Sung
4/17/22

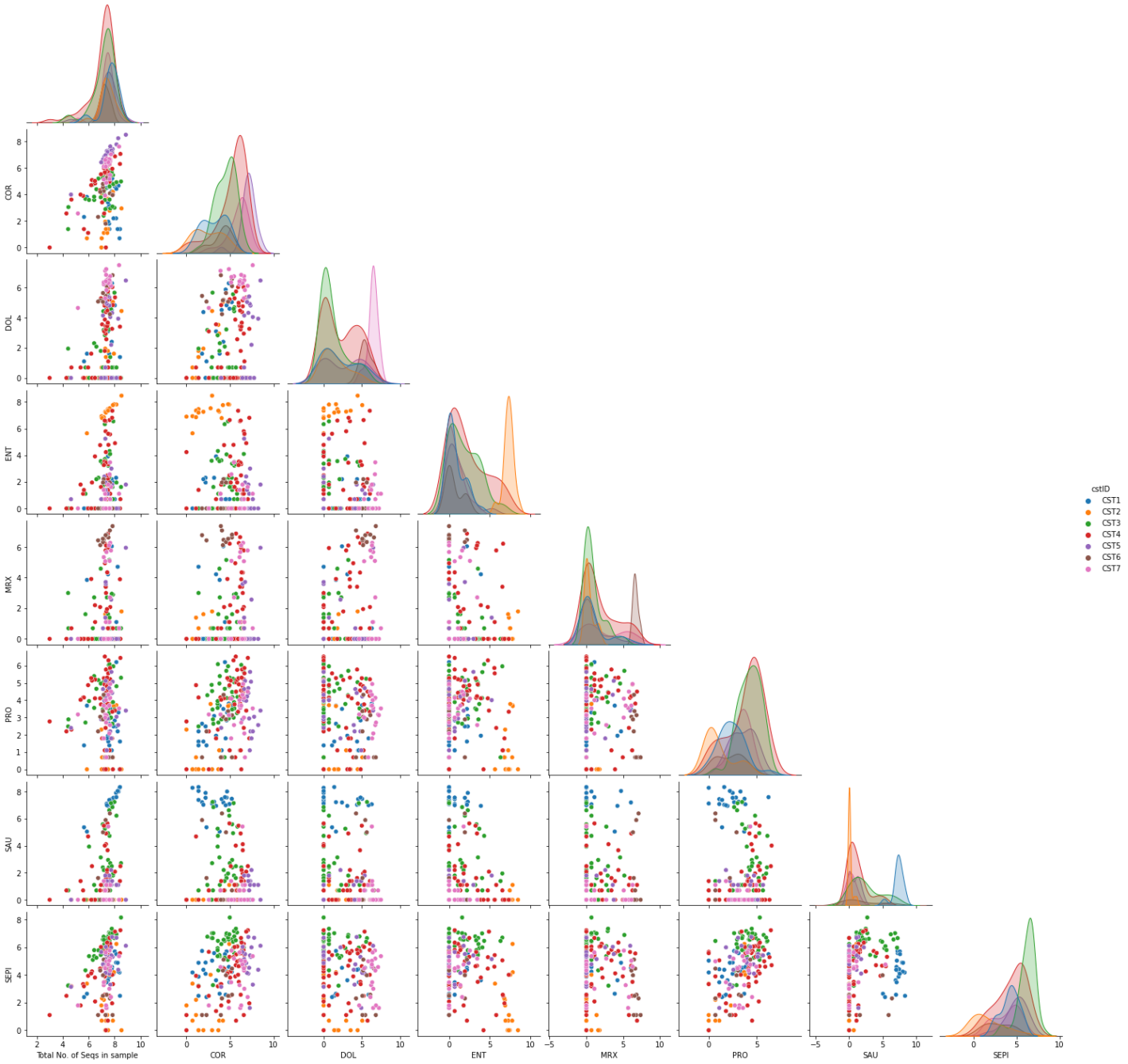
Histogram and KDE plots that show similar results seen in the clustered boxplots. Majority of the values cluster around the low end, with certain cst# appear on the high end depending on the feature bacteria group.



ARAC Lab Interview Problem Task

Edward Sung
4/17/22

Pairplot using log transformed data and clustering groups. Further analysis could be to separate the clusters to their own individual pairplots to clearly see the correlations. But certain correlations and separations can still be seen like combining SEPI and COR, where CST3 (Green) and CST4 (Red) are separated but positively correlated respectively.



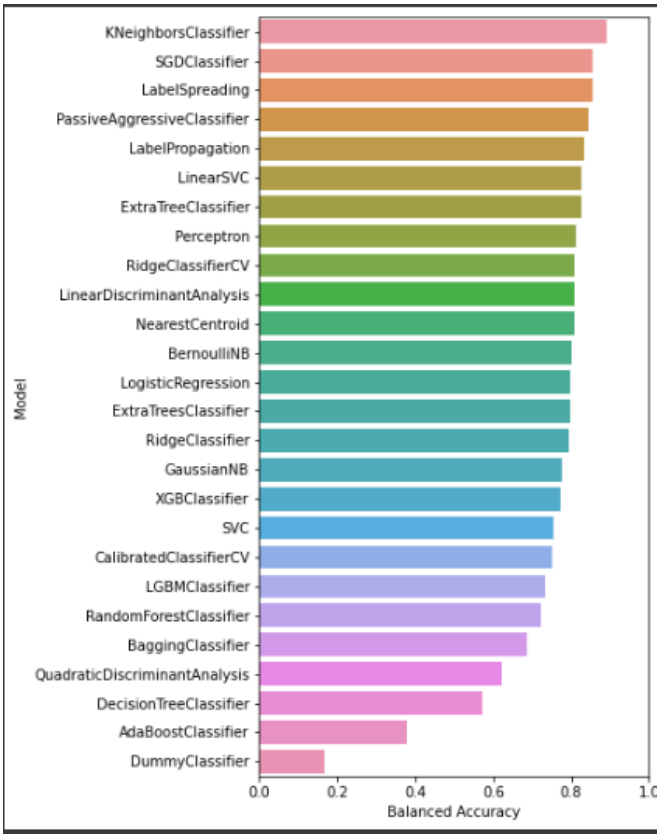
ARAC Lab Interview Problem Task

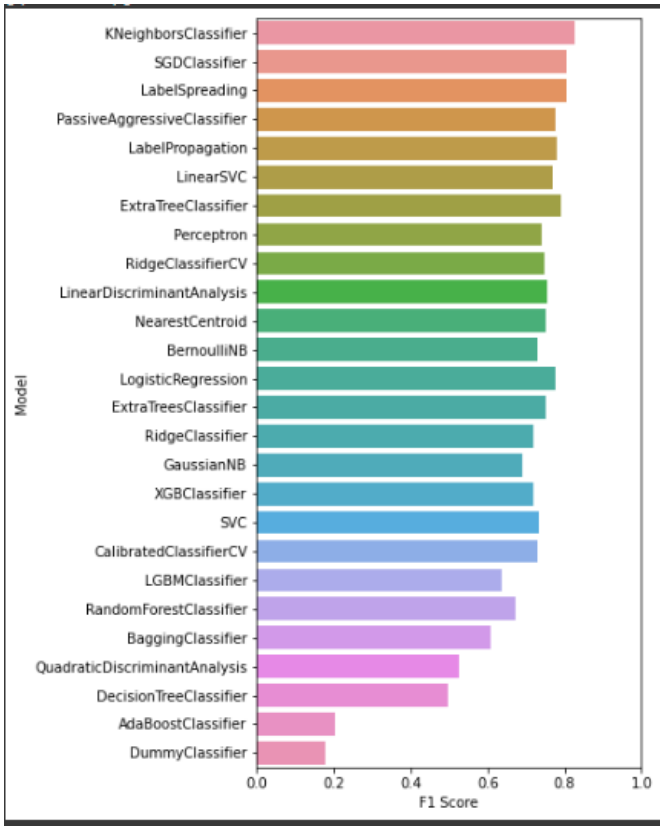
Edward Sung
4/17/22

Preprocessing and Modeling

I split the data using an 80/20 split, with 80% or 142 samples being part of the training set and 20% or 36 samples being part of the testing set. I also re-encoded the target column cstID since I converted it from its original one hot encoding to a single categorical column.

Using a package called lazypredict with a method called LazyClassifier, I ran 29 baseline models on the training data to see which model performs the best. I chose the model based on the balanced accuracy and F1 Score, since my model has some class imbalance to consider.





Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
KNeighborsClassifier	0.83	0.89	None	0.83	0.04
SGDClassifier	0.81	0.86	None	0.80	0.02
LabelSpreading	0.81	0.86	None	0.80	0.05
PassiveAggressiveClassifier	0.78	0.84	None	0.78	0.03
LabelPropagation	0.78	0.84	None	0.78	0.04

The top 5 base models are shown above, but I decided to follow through using SGDClassifer instead of KNeighborsClassifier. Even though both models are distance based and will be able to perform well on the data set as shown in the clustering EDA, where the same cluster points can be seen close to one another in the boxplot and in the pairplot, KNN suffers in general performance overtime since it needs to use historical database to calculate the nearest neighbors each time a prediction is needed versus a formula that can take an input and produce a prediction.

I also evaluated performing a power transformation given the skewness in the data set. But applying the box-cox and yeo-johnson transformation on the dataset and running it through the lazyclassifier produced a result with the highest Balanced Accuracy and F1 Score to be 0.77 and 0.72 respectively. This is a 0.9 and 0.8 drop respectively compared to not using a power transformation.

Following through with using SGDClassifier, I also evaluated and found that using the StandardScaler produced the best cross validation score on the base model. This aligns with what is said in the classifier documentation where it notes: “the data should have a zero mean and unit variance.” Note I used f1_weighted as my metric given the class imbalance and higher importance on predicting correctly rather than balanced accuracy, which considers both positive and negative classification.

Scaler	Mean CV Score	Standard Deviation
StandardScaler	78.9	0.21
MinMaxScaler	63.0	0.20
RobustScaler	53.6	0.24

I also applied class_weights=”balanced” to my SGDClassifier since it performed slightly better (~1%) than not including the weights.

Performing hyperparameter tuning using grid search identified my best parameters to be:

- Alpha = 0.01
- Loss = squared_hinge
- Max_iter = 100
- Penalty = None

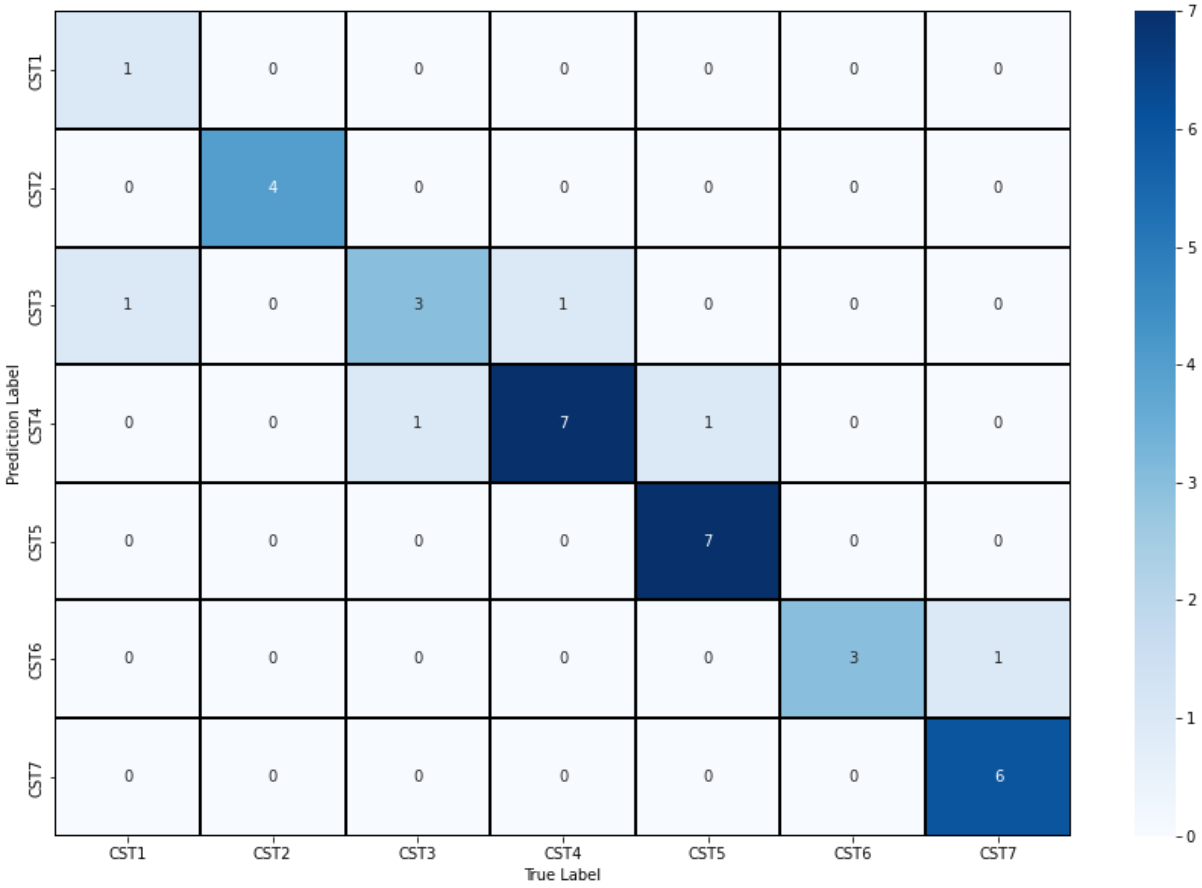
With a f1_weighted score of 0.875.

ARAC Lab Interview Problem Task

Edward Sung
4/17/22

Applying my model to the test set, I had an 86% accuracy with 5 misclassifications.

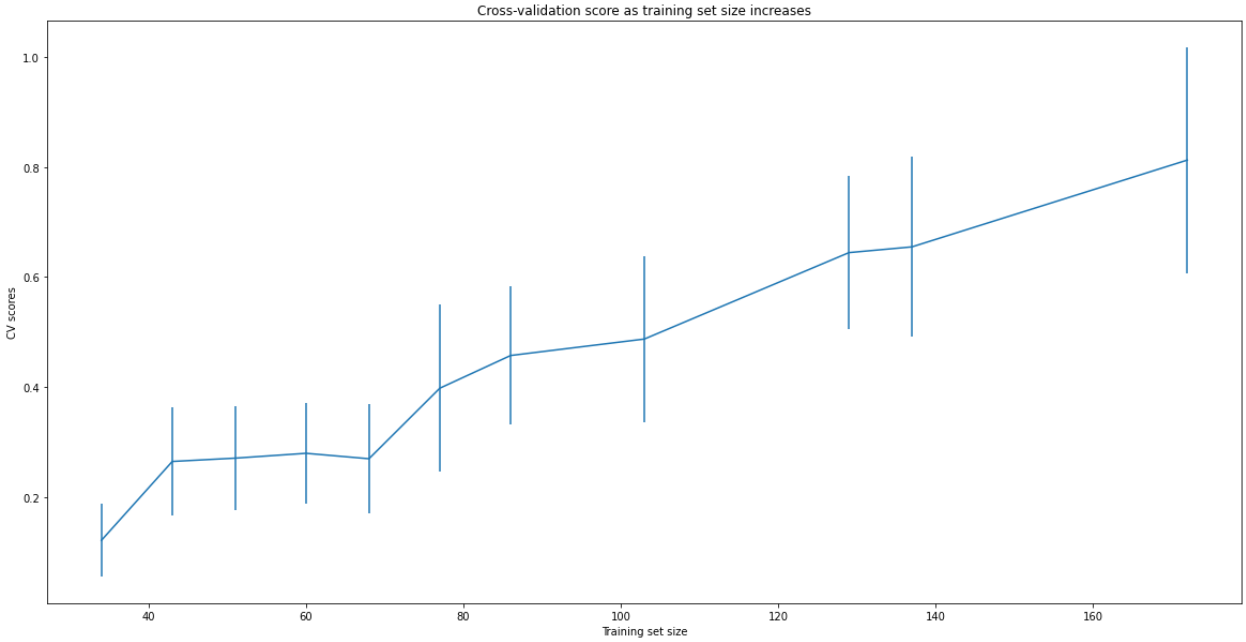
	precision	recall	f1-score	support
CST1	0.50	1.00	0.67	1
CST2	1.00	1.00	1.00	4
CST3	0.75	0.60	0.67	5
CST4	0.88	0.78	0.82	9
CST5	0.88	1.00	0.93	7
CST6	1.00	0.75	0.86	4
CST7	0.86	1.00	0.92	6
accuracy			0.86	36
macro avg	0.84	0.88	0.84	36
weighted avg	0.87	0.86	0.86	36



ARAC Lab Interview Problem Task

Edward Sung
4/17/22

I believe a large portion of the cause of the misclassification is lack of data and features. The graph below shows that with more data points, there is still plenty of room for the model to grow and improve. The low number of features is an issue as seen in the clustering boxplots. Example is CST1, where it can be visually seen that SAU can solely be used and have a large impact in classifying CST1. But if a value falls on the very low end of SAU, as seen in its lower whisker end point, it can be misclassified. A follow-up project could be to perform feature engineering to contribute more features to separate the clusters. More data will also help with the class imbalance as mentioned previously.



I have saved the model as a pickle that can be unpacked in another machine for usage, or I can follow-up with a project to deploy the model online as an application using Streamlit.