Charles Wilmot
Trey Eckenrod
Whiskey Data
Intro to ML and AI

“Optimizing Model Selection in Low Variability Datasets with Imbalanced Dependent Variables”

I. Abstract

This study aimed to determine the most accurate model for predicting data with low variability and an imbalanced dependent variable. Specifically, we focused on classifying scotch based on given features from the dataset. Four models were applied, namely KNeighbors, DecisionTree, RandomForest, and SVC, and it was found that the RandomForest model provided the best and most accurate prediction. This research is significant in exploring data analysis options when dealing with such constraints and parameters.When a data set has a large number of one dependent variable and a small number of the rest (imbalanced dependent variables) and most of the data is 0, meaning no to that specific feature, (low variability) we must find the best model to use when those are your certain constraints and parameters. It contributes to the field by exploring options of data analysis when you are given that sort of data.

II. Introduction

Accurately classifying and predicting data is critical in a highly data-driven world. However, when datasets have limited variability and one dependent variable has more data points than the others, determining the most appropriate model for analysis becomes a challenge. This study aimed to address this issue and determine the best model for analyzing data with uneven data points and low variability. We applied four models, namely KNeighbors, DecisionTree, RandomForest, and SVC, and evaluated their accuracy for classifying and predicting scotch based on given features from the dataset. RandomForest is the best model to use when you have that limited variability and one dependent variable has more data points than the rest. A question we asked was: “Which model is best for data with uneven data points and low variability?” It is important to answer this question because it will allow for better data analysis in the future when these parameters are met.

III. Background

Scotch was chosen as the target variable for this study and knowing about Scotch allowed us to better understand and utilize the dataset. Scotch is a low variability dataset because as the name suggests, all scotch is made in Scotland, a smaller region taking up only about one third of land on the island of Great Britain. The final product of a scotch has many smaller pieces that combine to create the whisky many people love, one of these pieces being the region of Scotland that it was distilled at. The reason for the variability in scotches can be attributed to historically how the different regions distilled it, as well as the temperature and humidity of different parts of the island, which could alter the final product after the long aging process. Despite using where the scotch was made as the dependent variable, the accuracy of the models used in this study was only around 80%. Therefore, we aimed to improve this metric to enable more accurate predictions, regardless of the type or amount of data provided. The data we worked on consisted predominantly of 1's and 0's with the ending columns consisting of columns we did not put into our model such as age, abv, a tasting score, and additional information on the smaller subcategories of the region in which the whisky is from. We chose to categorize by region as it provided us with a better chance of correctly categorizing scotches with our model. The smaller subcategories have as low as 1 scotch categorized under that district, which would prove to be very difficult to accurately categorize a low variance dataset correctly when trained on a single sample for some categories.

IV. Methods

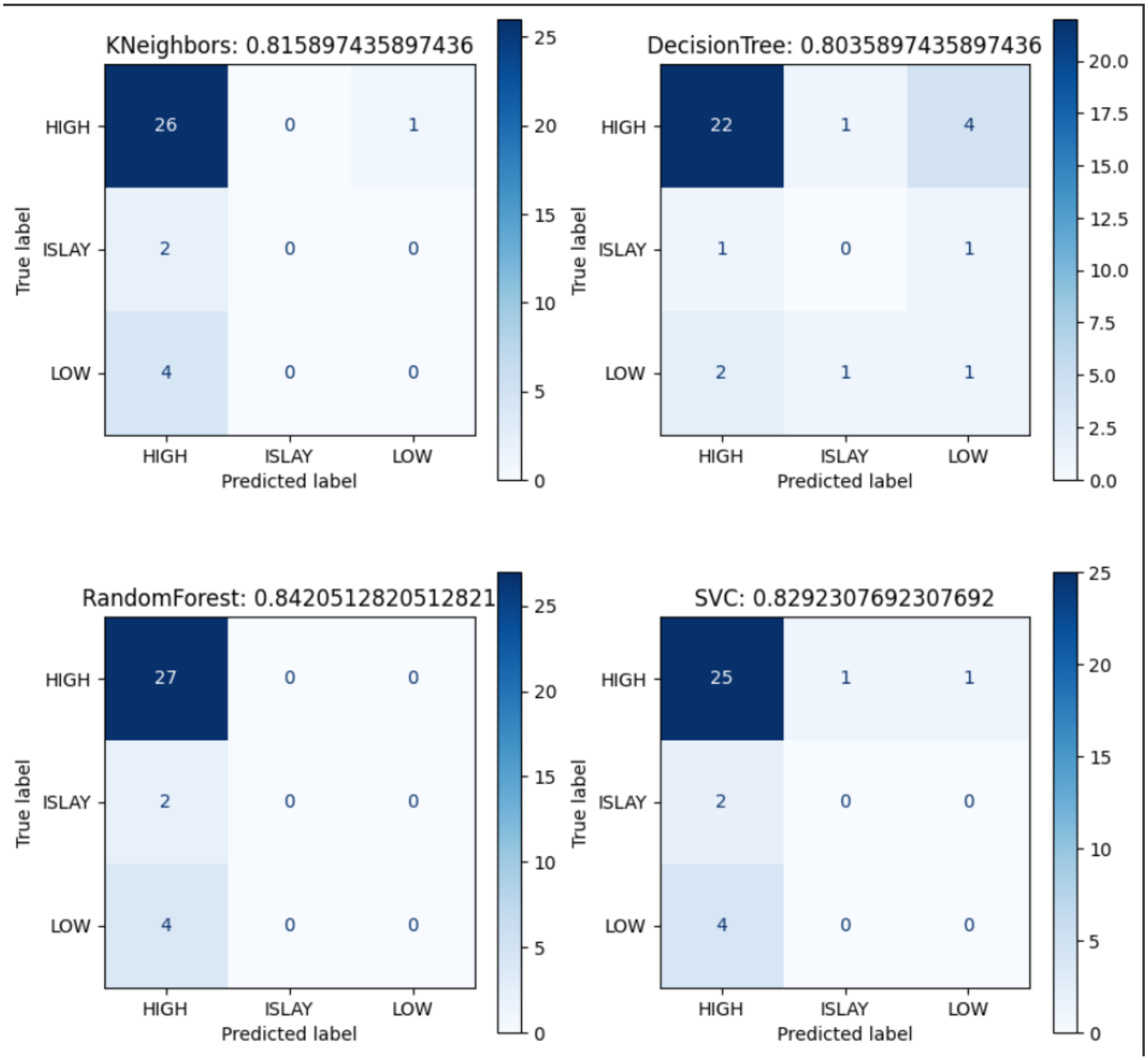
The four methods we used to find the best model are as follows. **KNeighbors**: K-Nearest Neighbors (KNN) is a simple algorithm that classifies new data points based on the majority class of their k-nearest neighbors in the training data. The value of k is a hyperparameter that needs to be set beforehand. **DecisionTree**: Decision trees are a type of supervised learning algorithm that can be used for both classification and regression tasks. The algorithm builds a tree-like model of decisions and their possible consequences. Each internal node of the tree corresponds to a decision on a feature, and each leaf node corresponds to a class label or a numerical value. **RandomForest**: Random forests are an ensemble learning method that combines multiple decision trees to improve the accuracy of the model. It works by creating a set of decision trees at training time and outputting the class that is the mode of the classes (classification) or the mean prediction (regression) of the individual trees. **SVC**: Support Vector Machines (SVM) is a popular algorithm for binary classification, but can also be used for multi-class classification and regression. The algorithm creates a hyperplane or set of hyperplanes in a high-dimensional space that separates the different classes. The goal of the algorithm is to find the hyperplane(s) that maximize the margin between the classes.

We collected the data by reading in an excel sheet and turning it into a numpy dataframe. We had to clean one data point that contained an extra space at the end by indexing it. We set X equal to the df.iloc[:, 1:69], meaning all the rows but only columns 1 through 69 and y equal to df.iloc[:, 72] which is all the rows but just the 72 column. We then did a label encoder on y and set y equal to the transform of the label encoder and performed train_test_split to get X_train, X_test, y_train, and y_test. We then made four different models: KNeighborsClassifier, DecisionTreeClassifier, RandomForestClassifier, and SVC. This was appropriate because we were trying to find the best classifier model to most accurately predict the dependent variable under these circumstances. Once we found the best model, RandomForest, we applied different scoring methods to it, chose the best, and visualized our results.

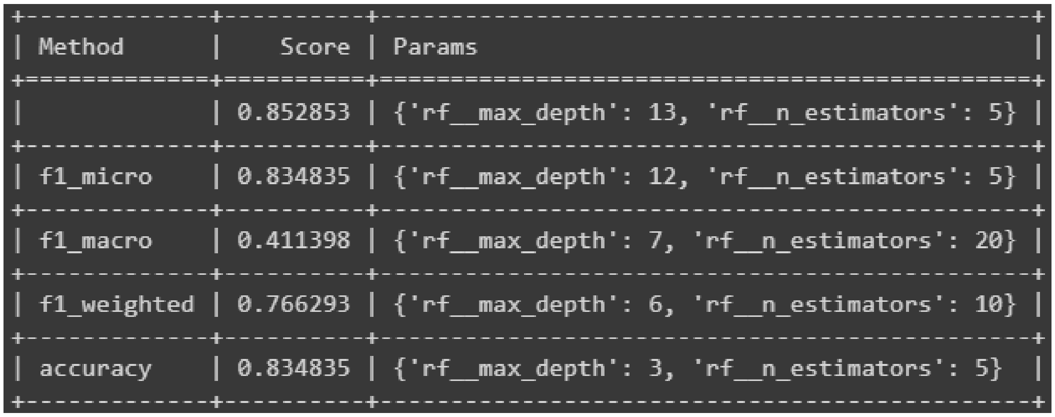
V. Results

The RandomForest model provided the best and most accurate predictions. The research process involved finding a dataset, evaluating different models, training the selected model, testing the model, visualizing the results, and displaying our findings.

CLF with Train, Train with Train, and Test with Test shows RandomForest to be the best score at roughly 84%.



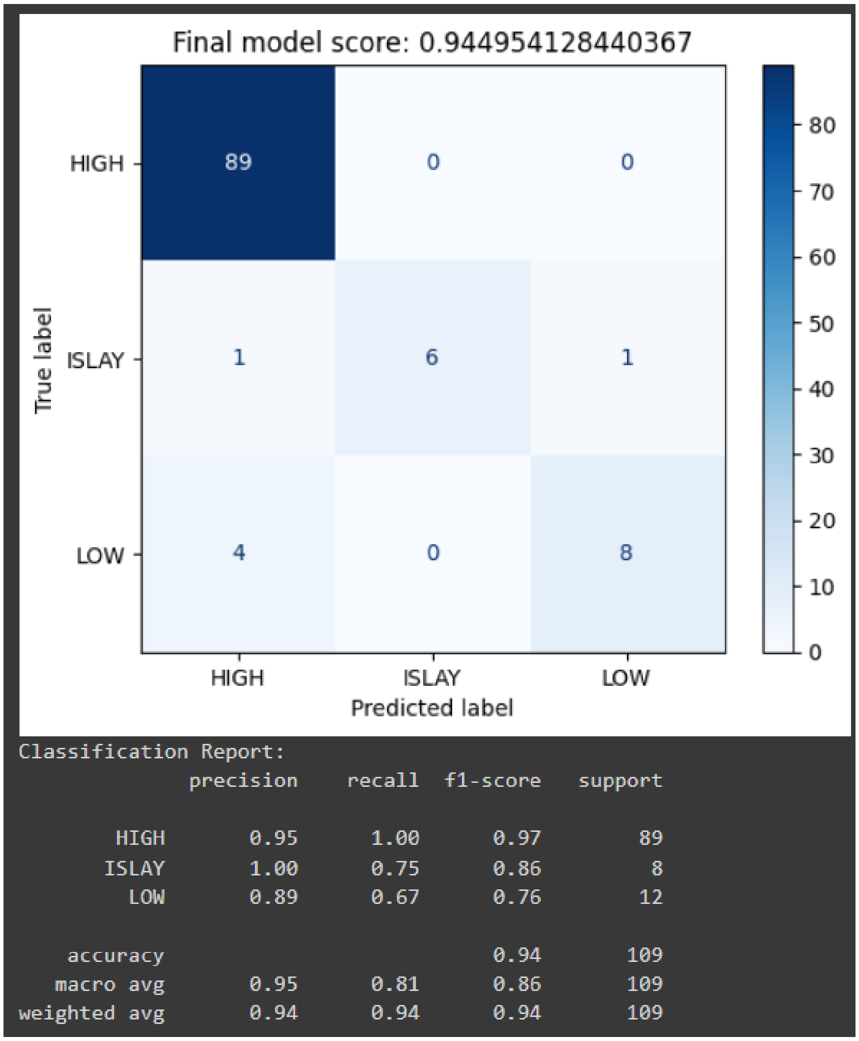
Then once we chose RandomForest we wanted to test how it did with other scoring methods:



```
+-------------+----------+--------------------------------------------+
| Method      |    Score | Params                                     |
+=============+==========+============================================+
|             | 0.852853 | {'rf__max_depth': 13, 'rf__n_estimators': 5} |
+-------------+----------+--------------------------------------------+
| f1_micro    | 0.834835 | {'rf__max_depth': 12, 'rf__n_estimators': 5} |
+-------------+----------+--------------------------------------------+
| f1_macro    | 0.411398 | {'rf__max_depth': 7, 'rf__n_estimators': 20} |
+-------------+----------+--------------------------------------------+
| f1_weighted | 0.766293 | {'rf__max_depth': 6, 'rf__n_estimators': 10} |
+-------------+----------+--------------------------------------------+
| accuracy    | 0.834835 | {'rf__max_depth': 3, 'rf__n_estimators': 5}  |
+-------------+----------+--------------------------------------------+
```

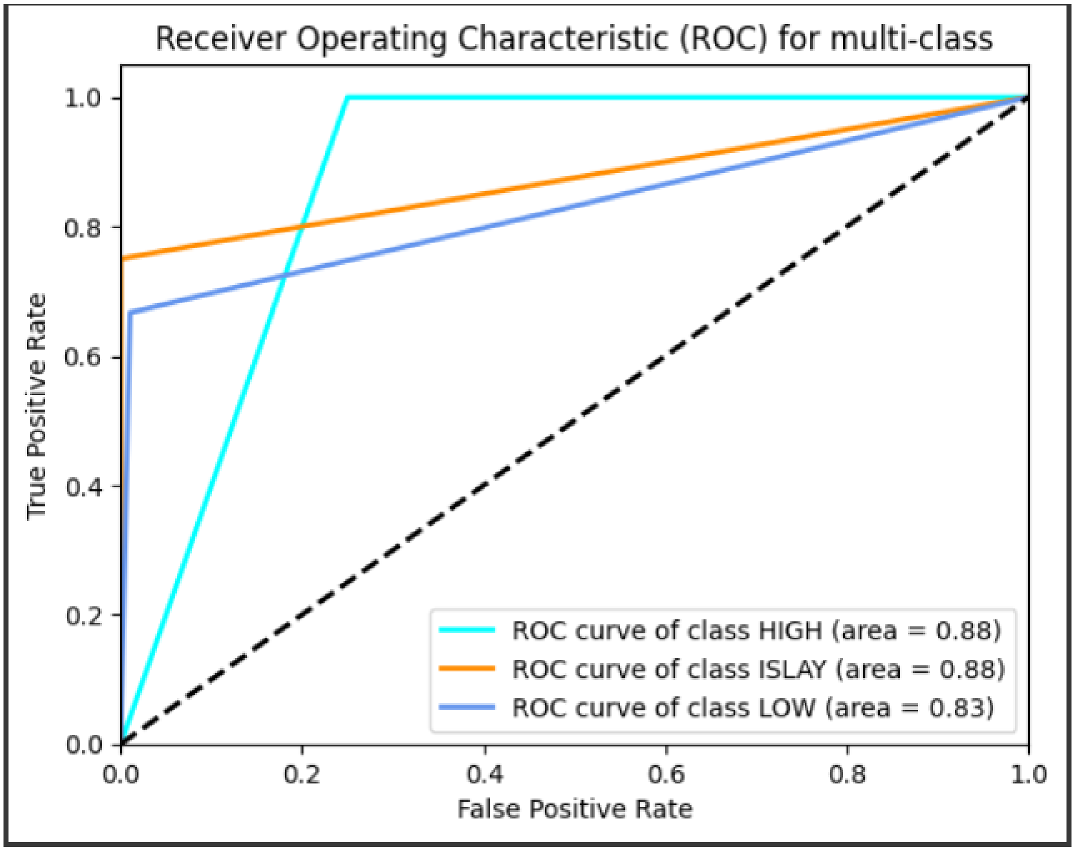
We see that the standard method did the best at 85% with the best parameters being max_depth of 13 and n_estimators of 5.

Then we constructed a Final Model Confusion Matrix:

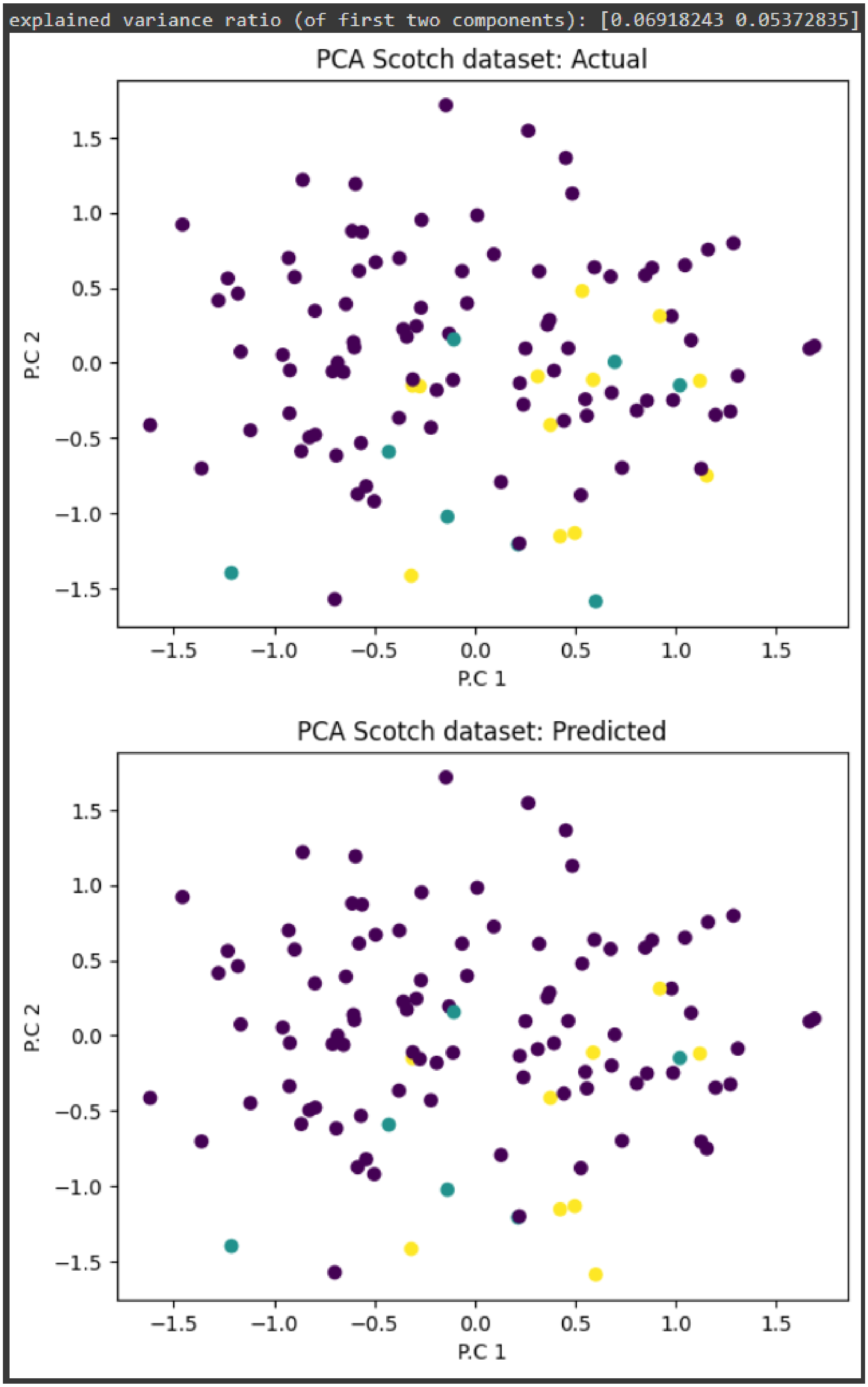


```
Classification Report:
              precision    recall  f1-score   support

        HIGH       0.95      1.00      0.97        89
       ISLAY       1.00      0.75      0.86         8
         LOW       0.89      0.67      0.76        12

    accuracy                           0.94       109
   macro avg       0.95      0.81      0.86       109
weighted avg       0.94      0.94      0.94       109
```

A ROC for multi-class graph:



And a PCA for the Actual data and the Predicted:



```
explained variance ratio (of first two components): [0.06918243 0.05372835]
```

VI. Conclusion

This study contributes to the field by providing insights into the analysis of data with limited variability and an imbalanced dependent variable. Our findings can help researchers determine the most appropriate approach when dealing with similar datasets. Some challenges we faced were cleaning of the data, working with data that has low variability and heavy imbalance of dependent variables, and getting meaningful results. A limitation was the amount of time it took to run through these models looking for the best. In future research, we suggest exploring more sophisticated techniques to improve accuracy and reduce the processing time for similar datasets.

VII. References

- Numpy https://numpy.org/
- Pandas https://pandas.pydata.org/
- Matplotlib https://matplotlib.org/
- Seaborn https://seaborn.pydata.org/
- Tabulate https://pypi.org/project/tabulate/
- Itertools https://docs.python.org/3/library/itertools.html
- Gomes, Joe. Lectures and Python notebooks on machine learning, AI, and data science https://uiowa.instructure.com/courses/204566
- KNeighborsClassifier https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- RandomForestClassifier https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- DecisionTreeClassifier https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
- SVC https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
- scikitLearn Scoring methods https://scikit-learn.org/stable/modules/model_evaluation.html
- PCA https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
- ROC Curve https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html
- Scotch Whisky Data https://www.numericalecology.com/data/scotch.html
- Scotland facts https://www.britannica.com/place/Scotland