Introduction: In order to better organize machine learning workflows, pipelines can be used to make work quicker. In this exercise, two different data sets (Red Wine Quality Data Set and Brain Stroke Data Set) were used to setup two different pipelines. Within the first pipeline, a scaling, feature selection, and Gradient Boost Classifier algorithm were combined. In the second pipeline, a scaling, feature selection, and K Neighbors Classifier algorithm were combined. Both pipelines (1 and 2) were tested on both data sets (Red Wine Quality Data Set and Brain Stroke Data Set). The mean accuracy score of the train and test data was then collected. Here, the mean accuracy score describes the mean accuracy of the given data across all subsets (cv).

Dataset Description: The first data set worked with was the Red Wine Quality data set. In this data set, red wine quality is measured by twelve features including fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol, and quality. For each of these features, the number of observations was 1599. Regarding preprocessing, I checked for null values using a isnull().values.any() function. There were not any null values. In this exercise, hyperparameter op-

timization was performed to improve the predicting model. The feature chosen to predict was quality.

The second data set worked with was the Brain Stroke data set. In this data set, occurrence of a stroke is measured by eleven features including gender, age, heart disease, ever married, work type, residence type, average glucose level, BMI, smoking status, and presence of stroke. For each of these features, the number of observations was 4981. Regarding preprocessing, I checked for null values by observing the table produced by .describe and also by checking with isnull().values.any() function. There were not any null values, however five features were listed as objects for data type. To change these to int, I used one hot encoding. The features changed from object to int using one hot encoding were gender, ever married, work type, residence type, and smoking status.

Experimental Setup: For this exercise, I used the Python programming language via Google Colab. For packages, pandas was used for data manipulation. The sklearn package was used for further data processing such as train_test_split, StandardScaler, GradientBoostClassifier, Pipeline, VarianceThresh-

old, KNeighborsClassifier, SimpleImputer, and GridSearchCV.

Beginning with Red Wine Quality Data Set Pipeline 1, the pipeline was built to include StandardScaler, VarianceThreshold, and GradientBoostinClassifier. This pipeline was fit and the mean accuracy train and test score generated before a grid search was conducted. Then, a grid search was performed on the same pipeline with the number of cross validation subsets being equal to three. Hyperparameters tuned here included classifier_max_depth (range of [1, 2, 3, 4, 5]) and classifier_min_samples_leaf (range of [10, 20, 30]). Specifically, classifier_max_depth describes the depth of the tree and classifier_min_samples_leaf describes the number of samples needed to form a leaf. After the grid search, the new mean accuracy train and test score were generated. Next for the Red Wine Quality Data Set, Pipeline 2 was assessed. Here, the pipeline was built to include StandardScaler, VarianceThreshold, and KNeighborsClassifier. This pipeline was fit and the mean accuracy train and test score generated before a grid search was conducted. Then, a grid search was performed on the same pipeline with the number of cross validation subsets being equal to three. Hyperparameters tuned here included selector_threshold (range of [0, 0.001, 0.01]),

classifier_n_neighbors (range of [1, 3, 5, 7, 10]), classifier_p (range of [1, 2]), and classifier_leaf_size (range of [1, 5, 10, 15]). Specifically, selector_threshold describes the cut off point for selection, classifier_n_neighbors describes the number of neighbors, classifier_p describes the power value for for the Minkowski metric, and classifier_leaf_size describes the number of samples in each leaf. After the grid search, the new mean accuracy train and test score were generated.

For the second data set, Brain Stroke Data Set, the same pipelines tested for the Red Wine Quality Data Set were tested with the same settings.

Results: Beginning with the Red Wine Quality Data Set Pipeline 1, before the grid search, the train and test mean accuracy scores were 0.8874120407 and 0.659375 respectively (Table 1). After grid search the train and test scores were 0.998436278342455 and 0.68125 respectively (Table 1). For Red Wine Quality Data Set Pipeline 2, the train and test mean accuracy scores were 0.70992963252541 and 0.515625 respectively (Table 2). After grid search, the train and test scores were 1 and 0.575 respectively (Table 2).

Next for the Brain Stroke Data Set Pipeline 1, before the grid search, the train and test mean accuracy scores were 0.9588353414 and 0.9478435306 respectively (Table 3). After grid search the train and test scores were 0.9503012048 and 0.9498495486 respectively (Table 3). For Brain Stroke Data Set Pipeline 2, the train and test mean accuracy scores were 0.9515562249 and 0.9488465396 respectively (Table 4). After grid search the train and test scores were 0.9505522088 and 0.9498495486 respectively (Table 4). Out of the two pipelines tested, Pipeline 1 saw the most improvement of mean accuracy test scores between both data sets. Because of this, pipeline 1 in this case was better setup for the data sets chosen.

Table 1: Red Wine Quality Data Pipeline 1	
Before Grid Search Train Score	0.8874120407
Before Grid Search Test Score	0.659375
After Grid Search Train Score	0.9984362783
After Grid Search Test Score	0.68125

Table 2: Red Wine Quality Data Pipeline 2	
Before Grid Search Train Score	0.7099296325
Before Grid Search Test Score	0.515625
After Grid Search Train Score	1
After Grid Search Test Score	0.575

Table 3: Brain Stroke Data Pipeline 1	
Before Grid Search Train Score	0.9588353414
Before Grid Search Test Score	0.9478435306
After Grid Search Train Score	0.9503012048
After Grid Search Test Score	0.9498495486

Table 4: Brain Stroke Data Pipeline 2	
Before Grid Search Train Score	0.9515562249
Before Grid Search Test Score	0.9488465396
After Grid Search Train Score	0.9505522088
After Grid Search Test Score	0.9498495486

References:

1. Saeed, Mehreen. "Modeling Pipeline Optimization with Scikit-Learn." MachineLearningMastery.Com, 21 Oct. 2021, machinelearningmastery.com/modeling-pipeline-optimization-with-scikit-learn/.

2. "Sklearn.Ensemble.Gradientboostingclassifier." Scikit, scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html. Ac-

cessed 3 Apr. 2024.

3. "Sklearn.Neighbors.Kneighborsclassifier." Scikit, scikit-learn.org/stable/modules /generated/sklearn.neighbors.KNeighborsClassifier.html. Accessed 3 Apr. 2024.

4. "Sklearn.Pipeline.Pipeline." Scikit, scikit-learn.org/stable/modules/generated /sklearn.pipeline.Pipeline.html. Accessed 3 Apr. 2024.