Wine Classification Problem

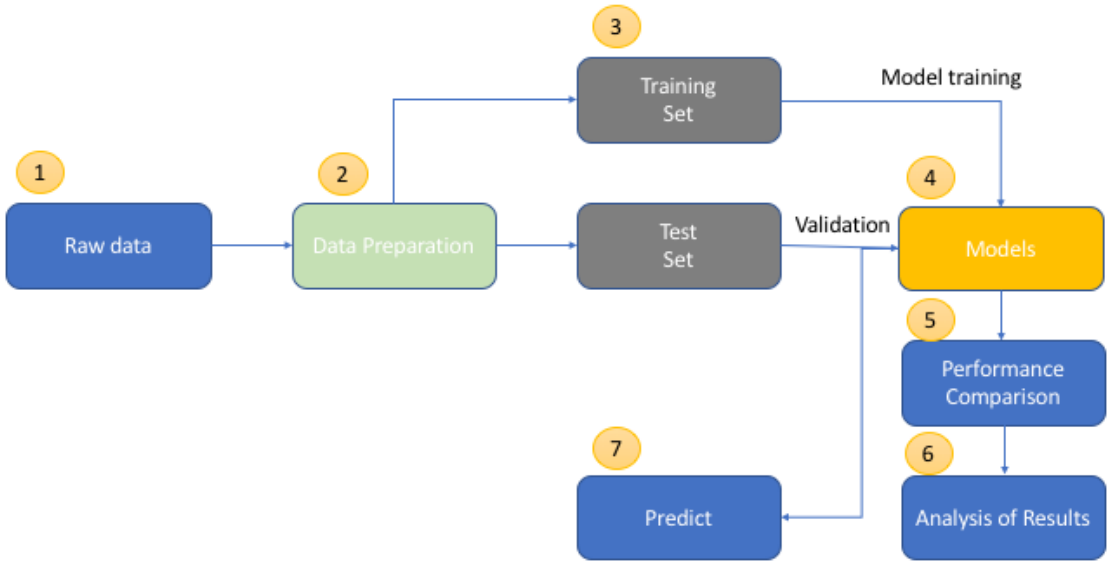
05/07/2019

Introduction

The intention of this project is to implement various classification algorithms to predict wine classes. Due to the complexity of wines based on their properties, we can let the training classifiers to tell us the categorical label for each wine along with several continuous-valued features.

This prototype will obtain the information from a small wine database in which is a copy of UCI ML Wine recognition datasets. I find that I am able to build different machine learning models and ultimately train them to compare their performance and accuracy during this prototype project.

Data Pipeline Architecture Overview



As part of the architecture overview, I will explain the Data pipeline building process that can be broken into several steps

1. Raw data

In this particular step, the main focus is to extract or gather the raw data. As part of this prototype, the initial phase consisted of loading the wine dataset from a python library called sklearn in which data is a copy of UCI ML Wine recognition datasets.

2. Data Preparation

As part of this step or phase, I divided the data into two small sub-tasks

2.1. Exploration and Sanitization

This step involved exploring and visualizing the data to ultimately map out the most interesting features within our data set. As part of the sanitization, it is imperative to remove all possible outliers or features that may potentially create bias in my models.

2.1.a The data exploration consisted in the following objectives:
Inspection of dataset characteristics such as number of instances, number of attributes, attribute information such as the column names, summary of statistics among the different wine classes

2.1.b Verification of possible target variables or labels. After the inspection of dataset characteristics, I found that the given dataset contains three main labels or classes in which I used in order to predict in my models.

2.1.c Data visualization was relevant as part of this project. In order to find anomalies and verify the relationship among features, I utilized seaborn library in order to see the distribution of observations and a gaussian kernel density estimate.

It is important to mention that the distribution of observation graph was be applied to features such as alcohol to understand its distribution among all the wines or observations and to possibly distinguish the alcohol content by classes.

Once the given information was visualized and I obtained some data insight about the alcohol distribution among wine classes, I applied the same approach across other interesting wine's features such as malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, non flavanoid phenols, proanthocyanins, color intensity, hue, proline

After observing all the possible features from the given dataset, I confirmed that all these various features and target variables are relevant enough to include or feed into my models. No anomalies or sanitization process was needed as part of this project

2.2 Transformation

This step involves transforming my data (normalization, translation, etc) before training my models.

In order to effectively train and test my models, I converted the raw data into data frames in which I will implemented Pandas library in order to hold and split my data into two different categories:

- **Feature data set:** it contains all the features being described as part of the exploration process

- **Target data set:** it contains the wine classes which consists of 3 types.

3. Train/Test Data split

For this, I used Scikit-learn library. The data split technique or approach being used is called the **Train Test split approach** which consists of randomly splitting the complete data into a proportions of 80:20 (80% belongs to the training set and 20% to the test set).

The idea behind the data split is to use the training set to train my models and the test set to evaluate or predict through the trained models

4. Model training

This phase consists of Model selection, training and validation.

For our classification models, I used Logistic Regression, K Nearest Neighbors, Linear SVM, Gradient Boosting, Decision Tree, Random Forest and Naïve Bayes Classifiers.

All these models should be able to capture most of the complexity of the training data and validation score will depend on the amount of training data. After training the models, the expectation is to see the behavior of every model with a generalized solution by using the testing data without any overfitting or underfitting issue.

It is also important to mention that I used a dictionary containing all the classifiers with instances of every single model by using the sklearn library.

5. Performance Comparison

After fitting and full training each model, then I calculated the accuracy of the predictions on the training and test sets to see how well the model generalizes to data that never has seen before. I sorted the models based on their predictions and their performance during the training process.

6. Analysis of Results

To analyze the results, I implemented multiple graphs which consists of plotting the given results from the performance comparison to see the train and test scores during the training time for each classifier.

After observing them, most of the classifiers performed well except for Nearest Neighbors.
Random Forests, Linear SVM and Gradient Boosting classifiers were able to perform reasonably well. They didn't show high bias in order to capture all of the nuance of the data.

7. Predict

We could say that Naïve Bayes, Random Forest, Gradient Boosting and Linear SVM can be good models for real world solutions specifically for wine classification tasks as they reliably predict wine classes with an accuracy greater than 95% of the time.

Data structures and Algorithms

General Data description

```
target_names
 ['class_0' 'class_1' 'class_2']

feature_names
 ['alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium',
'total_phenols', 'flavanoids', 'nonflavanoid_phenols', 'proanthocyanins',
'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline']
```

Data characteristics

```
    Number of Instances: 178 (50 in each of three classes)
    Number of Attributes: 13 numeric, predictive attributes and the class
    Attribute Information:
              - 1) Alcohol
              - 2) Malic acid
              - 3) Ash
              - 4) Alcalinity of ash
              - 5) Magnesium
              - 6) Total phenols
              - 7) Flavanoids
              - 8) Nonflavanoid phenols
              - 9) Proanthocyanins
              - 10)Color intensity
              - 11)Hue
              - 12)OD280/OD315 of diluted wines
```

```
- 13)Proline
- class:
  - class_0
  - class_1
  - class_2
```

Relevant Summary Statistics

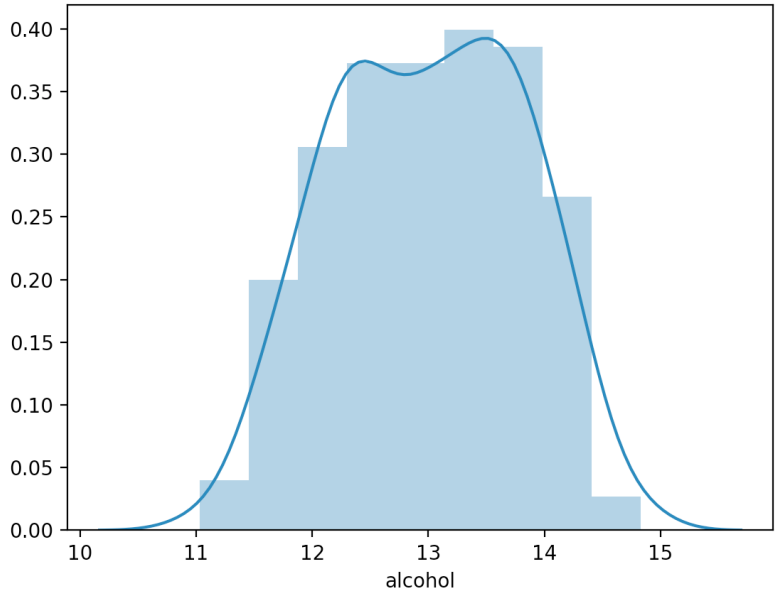
```
============================== ==== ===== ======= =====
                                Min   Max   Mean     SD
============================== ==== ===== ======= =====
Alcohol:                       11.0  14.8    13.0   0.8
Malic Acid:                    0.74  5.80    2.34  1.12
Ash:                           1.36  3.23    2.36  0.27
Alcalinity of Ash:             10.6  30.0    19.5   3.3
Magnesium:                     70.0 162.0    99.7  14.3
Total Phenols:                 0.98  3.88    2.29  0.63
Flavanoids:                    0.34  5.08    2.03  1.00
Nonflavanoid Phenols:          0.13  0.66    0.36  0.12
Proanthocyanins:               0.41  3.58    1.59  0.57
Colour Intensity:               1.3  13.0     5.1   2.3
Hue:                           0.48  1.71    0.96  0.23
OD280/OD315 of diluted wines: 1.27  4.00    2.61  0.71
Proline:                       278  1680     746   315
============================== ==== ===== ======= =====
```

Snapshot of the Dataset content

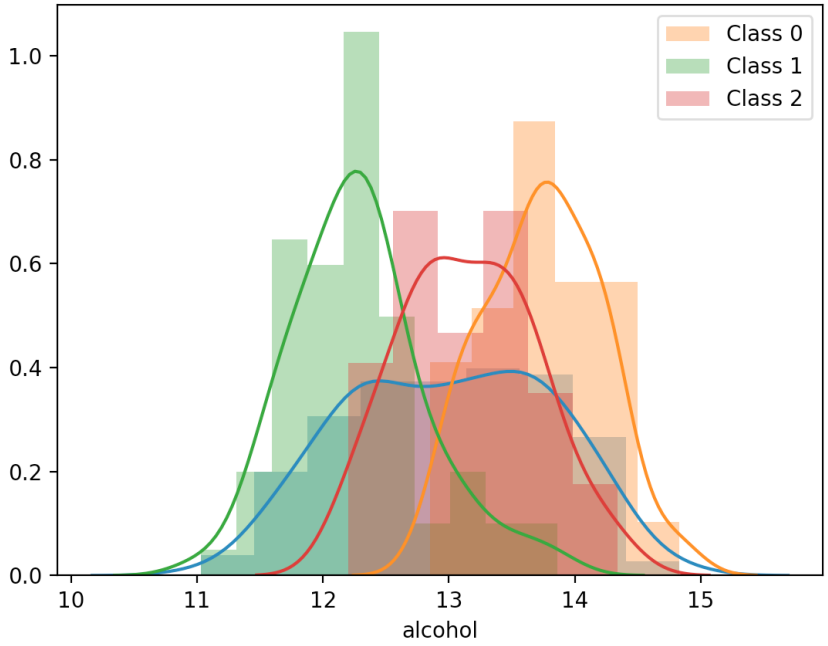
Alcohol	Malic acid	Ash	Alkalinity of ash	Magnesium	Total phenols	Flavanoids	Non-flavanoid phenols	Proanthocyanins	Color intensity	Hue	Diluted wines	Proline	Class
14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	Class 0
13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	Class 0

Experiments and Tests Results

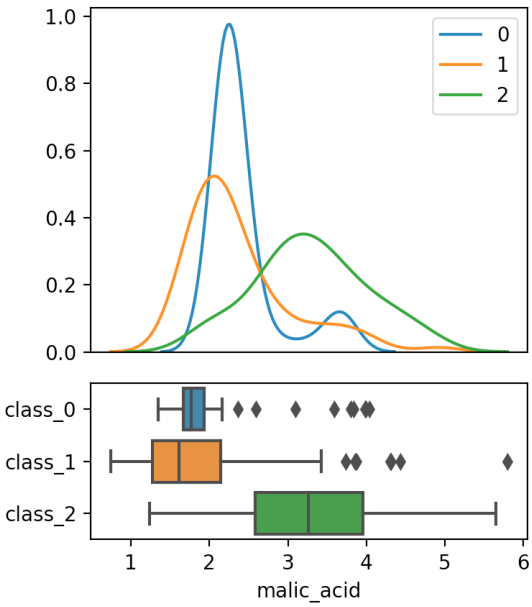
Alcohol content distribution among all wines

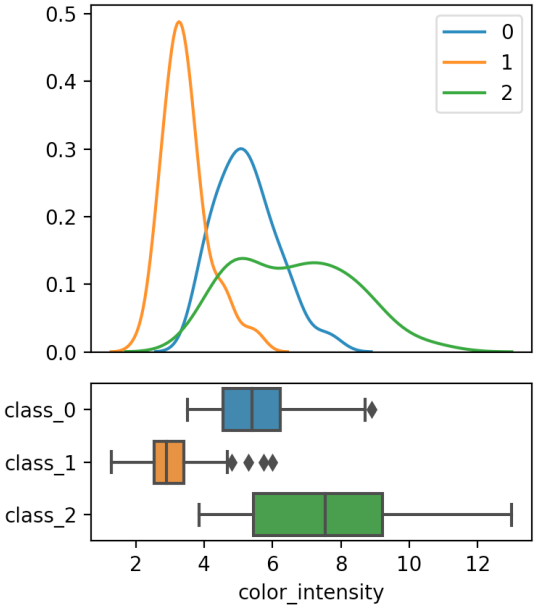


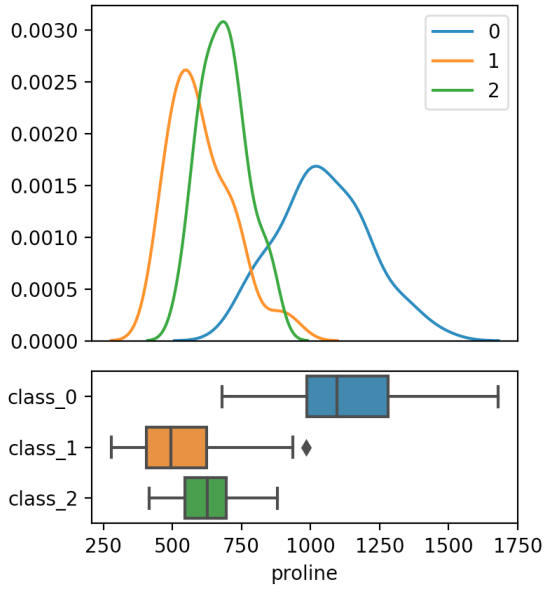
Alcohol content distribution among wine classes

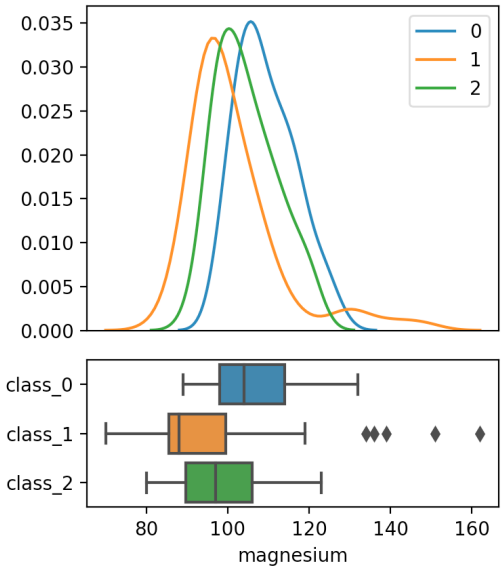


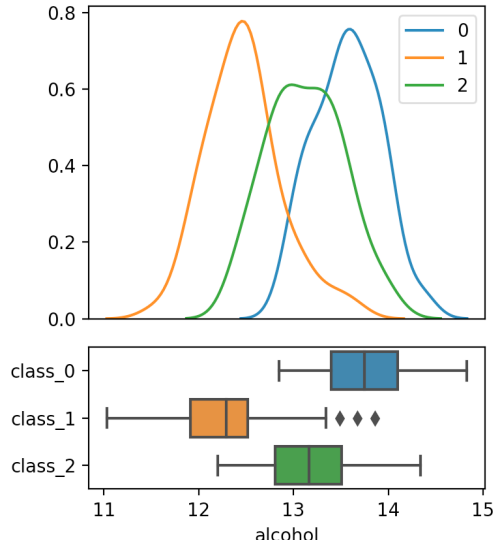
Features Gaussian Kernel Density Estimate among wine classes

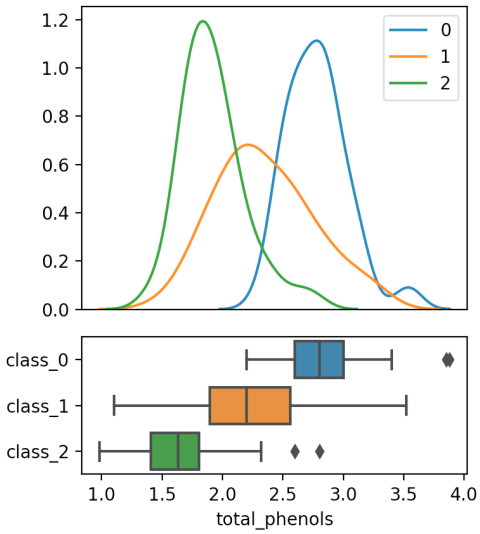


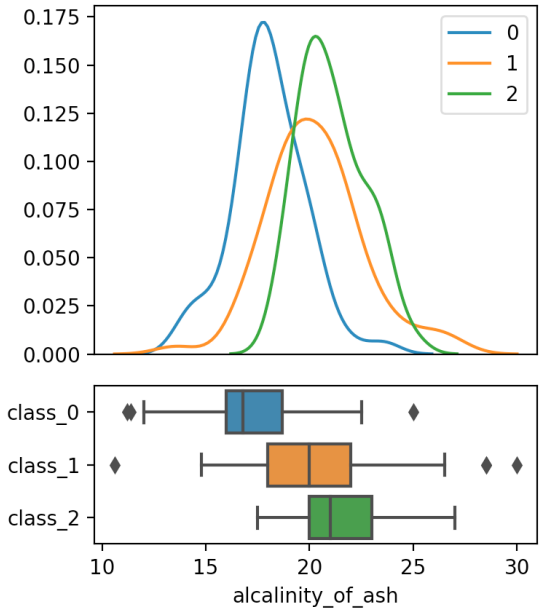


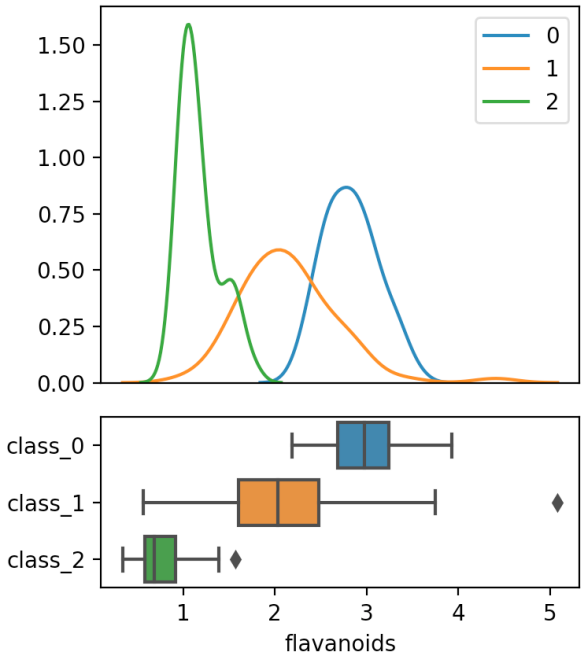


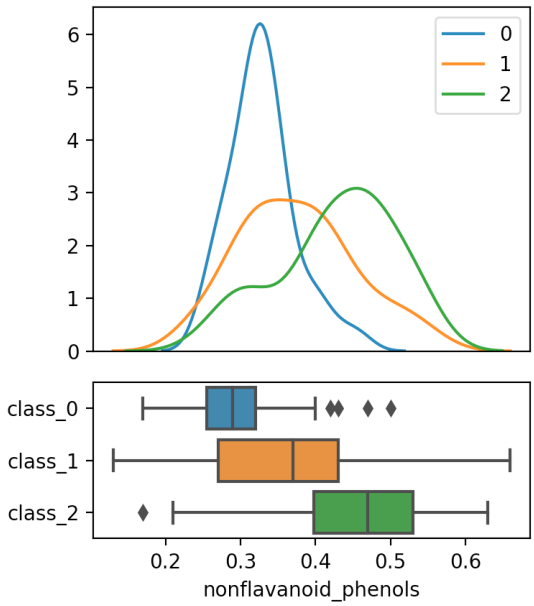


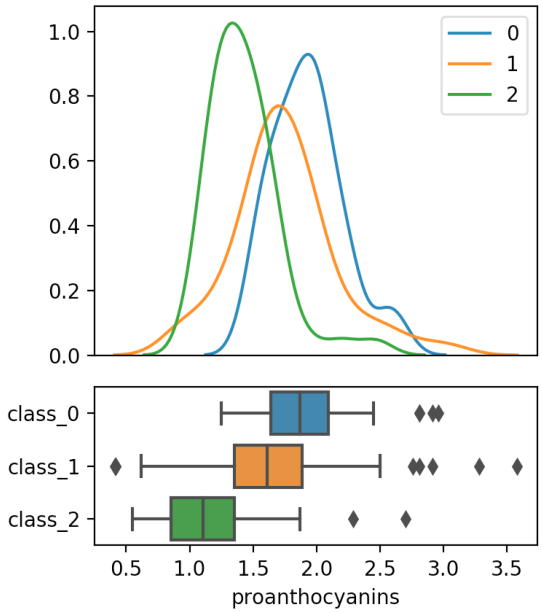

0
1
2
class_0
class_1
class_2
alcohol

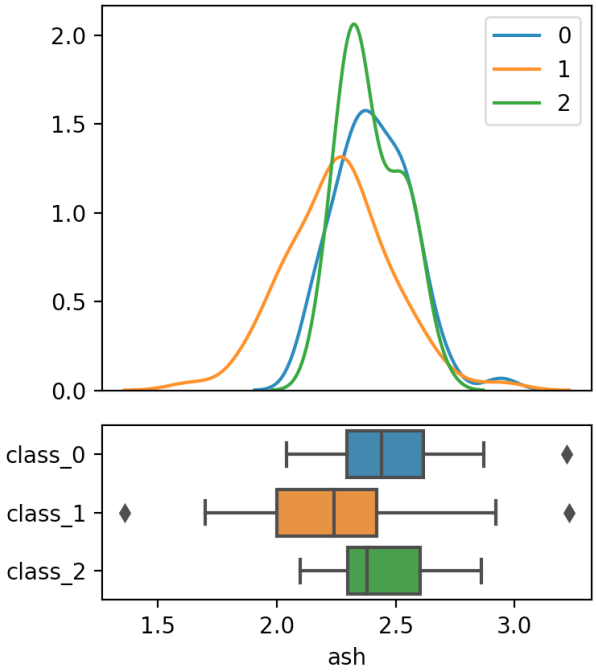

0
1
2
class_0
class_1
class_2
total_phenols

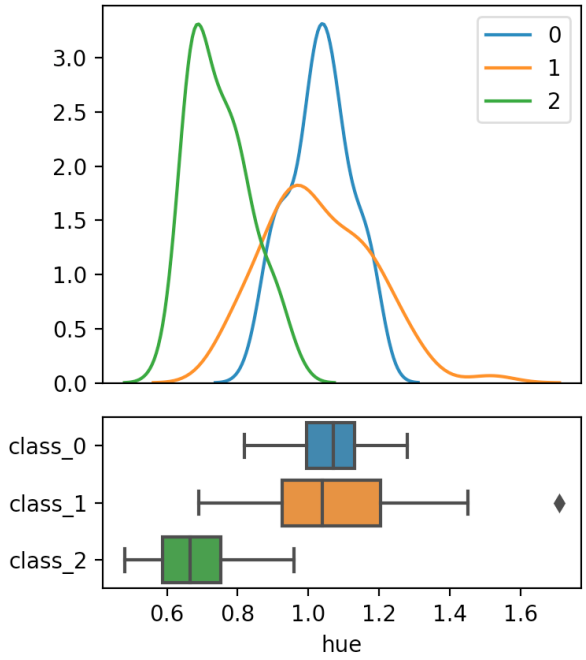

0
1
2
class_0
class_1
class_2
alcalinity_of_ash

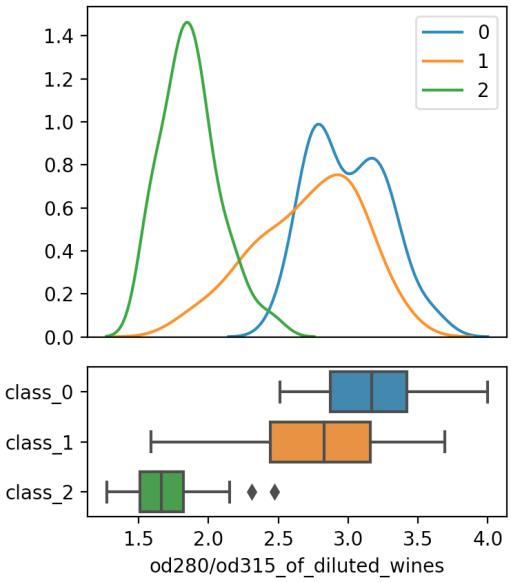

0
1
2
class_0
class_1
class_2
flavanoids


0
1
2
class_0
class_1
class_2
0.2
0.3
0.4
0.5
0.6
nonflavanoid_phenols


0
1
2
class_0
class_1
class_2
0.5
1.0
1.5
2.0
2.5
3.0
3.5
proanthocyanins


0
1
2
class_0
class_1
class_2
1.5
2.0
2.5
3.0
ash


0
1
2
class_0
class_1
class_2
0.6
0.8
1.0
1.2
1.4
1.6
hue

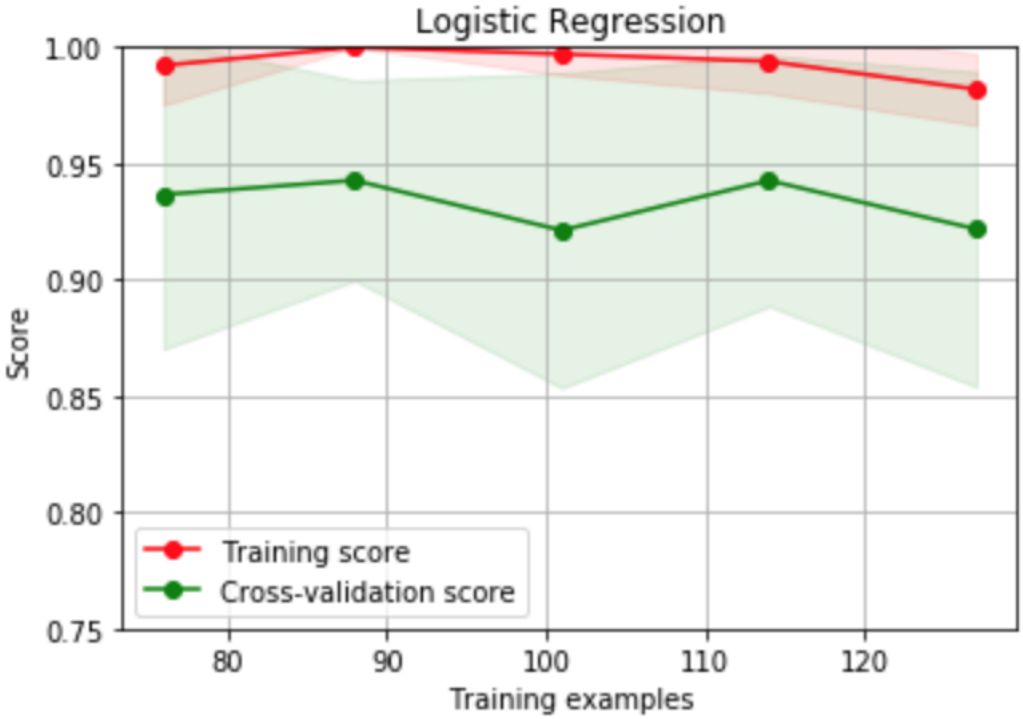


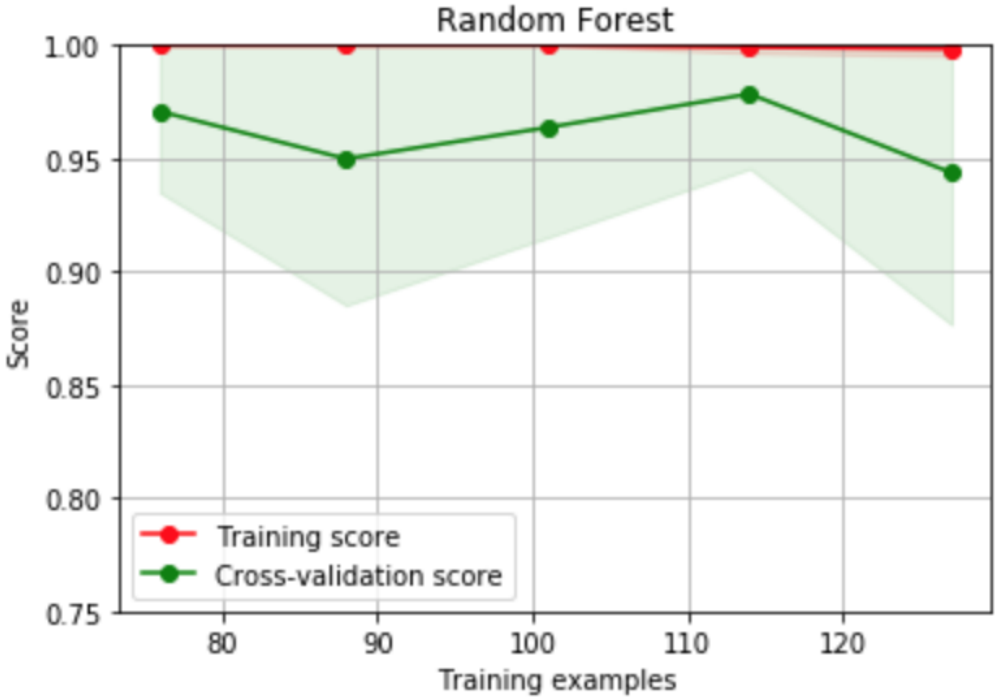
Classifiers Performance results

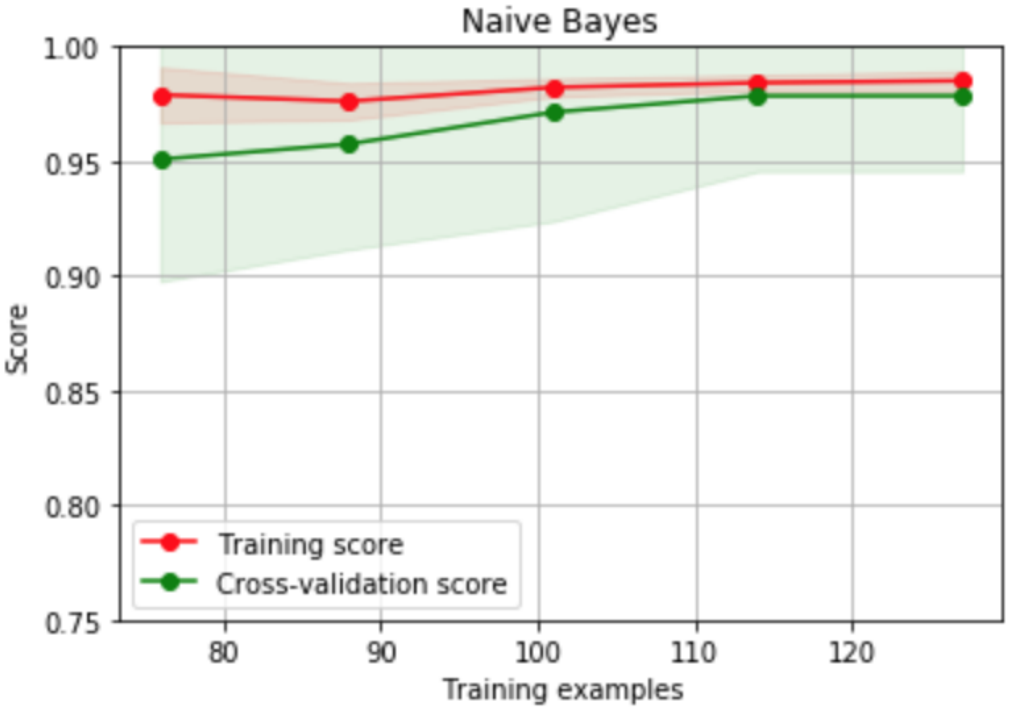
```
trained Logistic Regression in 0.36 s
trained Random Forest in 0.18 s
trained Naive Bayes in 0.14 s
trained Decision Tree in 0.18 s
trained Gradient Boosting Classifier in 0.84 s
trained Nearest Neighbors in 0.20 s
trained Linear SVM in 0.55 s
```

Classifiers Train and Test scores

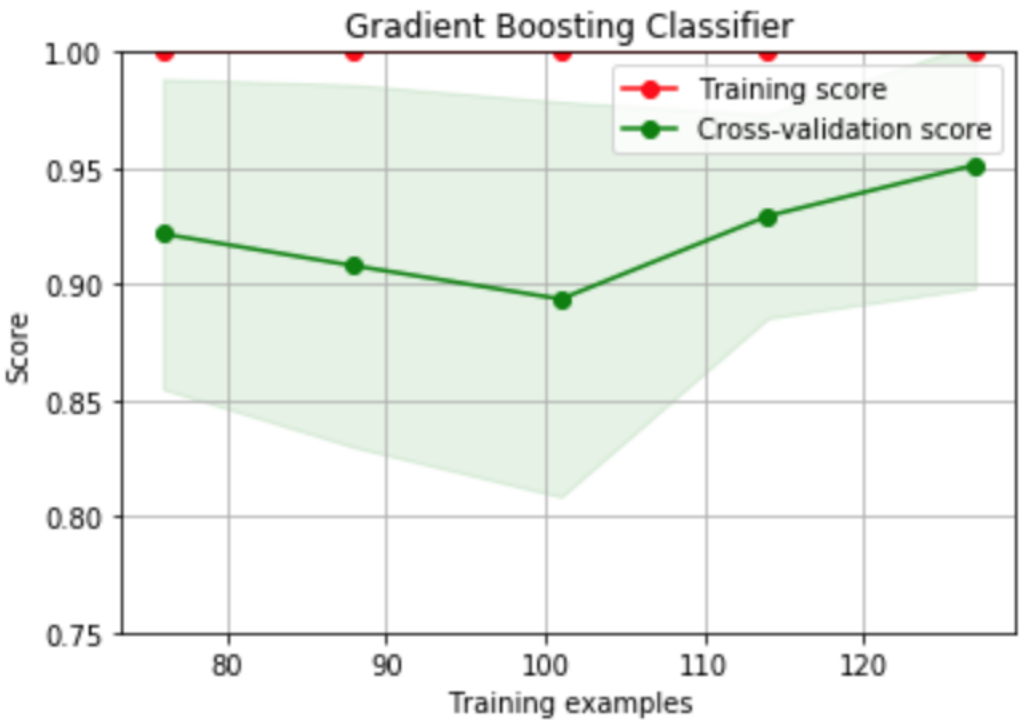
	classifier	train_score	test_score	training_time
0	Logistic Regression	0.964789	1.000000	0.356717
1	Random Forest	1.000000	1.000000	0.175839
2	Naive Bayes	0.985915	1.000000	0.137670
4	Gradient Boosting Classifier	1.000000	1.000000	0.836090
6	Linear SVM	1.000000	0.972222	0.551079
3	Decision Tree	1.000000	0.944444	0.181586
5	Nearest Neighbors	0.774648	0.777778	0.203932

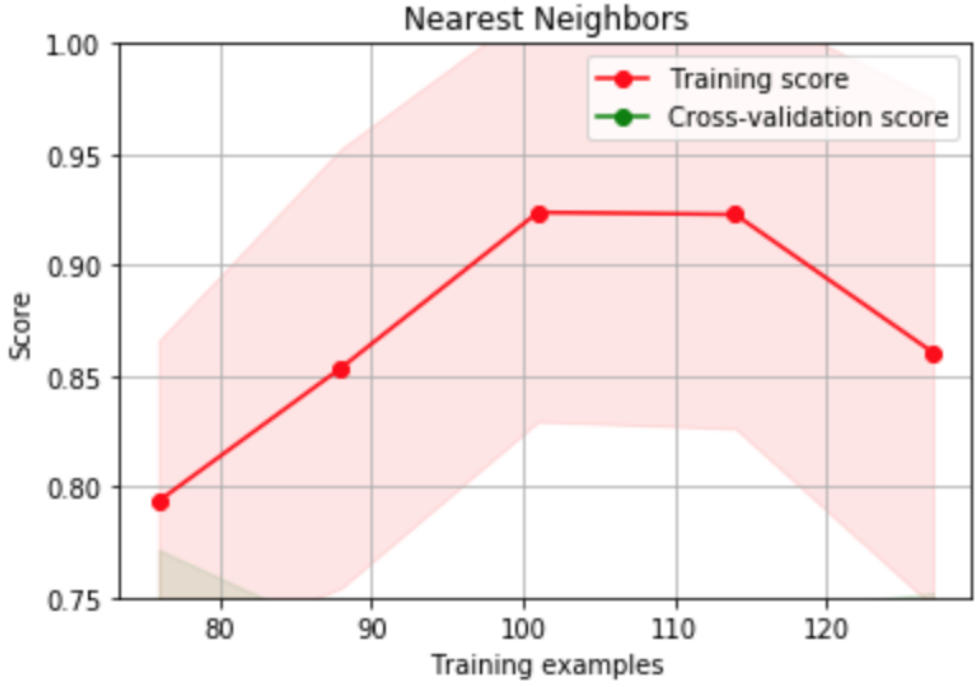

Logistic Regression
1.00
0.95
0.90
0.85
0.80
0.75
Score
Training score
Cross-validation score
80
90
100
110
120
Training examples

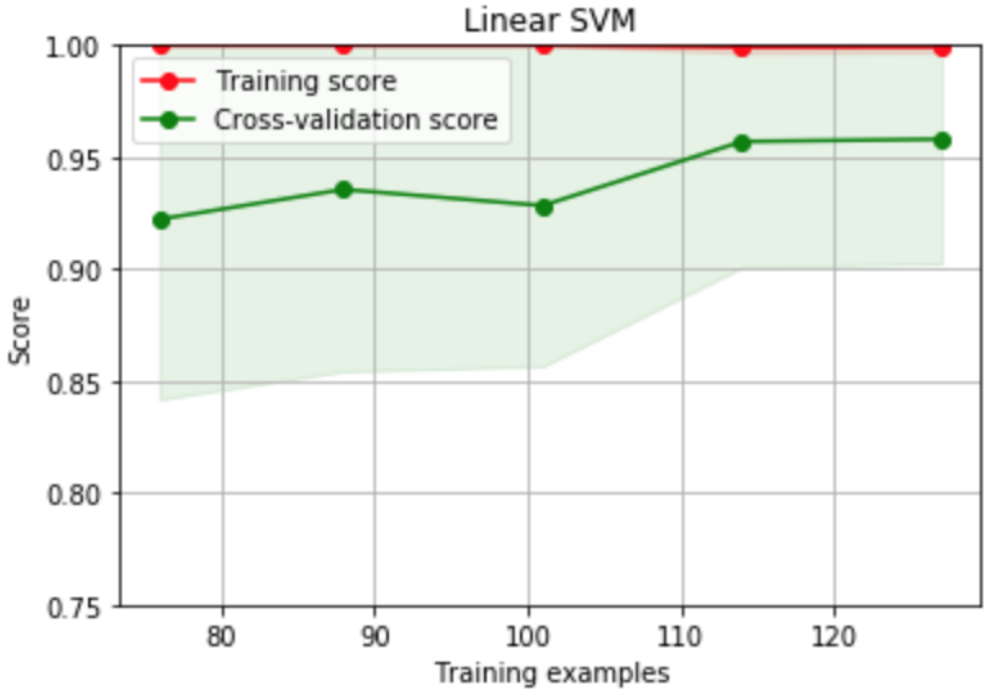

Random Forest
1.00
0.95
0.90
0.85
0.80
0.75
Score
Training score
Cross-validation score
80
90
100
110
120
Training examples


Naive Bayes
1.00
0.95
0.90
0.85
0.80
0.75
Score
80
90
100
110
120
Training examples
Training score
Cross-validation score


Decision Tree
1.00
0.95
0.90
0.85
0.80
0.75
Score
80
90
100
110
120
Training examples
Training score
Cross-validation score


Gradient Boosting Classifier
Training score
Cross-validation score
Score
Training examples
1.00
0.95
0.90
0.85
0.80
0.75
80
90
100
110
120


Nearest Neighbors
Training score
Cross-validation score
Score
Training examples
1.00
0.95
0.90
0.85
0.80
0.75
80
90
100
110
120



Discussion from similar projects

This Wine classification prototype has been inspired by similar projects. While this project has implemented a collection of classifiers by using supervised learning algorithms to classify the wine classes. I have created a table in which the same classification task or problem has been solved differently or similarly to my project. The following table will contain a list of ML algorithms in which I have found from other projects to implement

ML algorithms	**Type of Algorithm**
Logistic Regression	Supervised Learning
Random Forest	Supervised Learning
Naïve Bayes	Supervised Learning
Gradient Boosting Classifier	Supervised Learning
Linear SVM	Supervised Learning
Decision Tree	Supervised Learning
KNN	Supervised Learning
Neural Network	Supervised Learning
LSTM	Supervised Learning

Reference from similar projects

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6909240.pdf

https://www.mathworks.com/help/deeplearning/examples/wine-classification.html;jsessionid=3b6909c35117e5a9b3b77daf195f#:~:targetText=A%20neural%20network%20with%20enough,solving%20the%20wine%20classification%20problem.

https://jstevenr.com/wine-classification.html

https://www.freecodecamp.org/news/using-data-science-to-understand-what-makes-wine-taste-good-669b496c67ee/

https://towardsdatascience.com/wine-ratings-prediction-using-machine-learning-ce259832b321