# Project1_wine_classification

February 17, 2023

# 1 Wine Quality Classification

**Abstract**

Quality is the degree to which a product meets specified requirements. To assess white wine quality, it is essential to select measure(s) that directly impact its quality. In this study, we will be using physicochemical properties as features to evaluate the white wine's quality attribute.

**Background**

Our client is the retailer and wholesaler, Liquor Control Board of Ontario (LCBO). They would like to assess white wine quality to determine its prices as part of their research quality management. Wine quality can be assessed either by physicochemical properties or by human sensory testing. Physicochemical properties include pH, dissolved salts, sodium levels, the acidity, and density. As the demand of high-quality wine is increasing, the need for better prediction of wine quality in an efficient and convenient way is also in high demand. Human sensory testing of wine quality can be a time-consuming process and open to interpretation. Another method in wine informatics is exploring machine learning techniques to classify various wine attributes such as quality based on wine quality evaluation.

**Objective** The objective of this study is to use binary classification model to determine which white wines are high quality or low quality based on several important physicochemical properties. We use the white wine quality dataset retrieved from the UCI Machine learning repository: https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

**References**

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Mani, S., Krishnankutty, R. A., Swaminathan, S., & Theerthagiri, P. (2023). An investigation of wine quality testing using machine learning techniques. IAES International Journal of Artificial Intelligence, 12(2), 747.

## 1.1 Install Packages and Load in Dataset

[ ]:
```
!pip install pycaret
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
```

```
Collecting pycaret
  Downloading pycaret-2.3.10-py3-none-any.whl (320 kB)
                              320.2/320.2

KB 4.2 MB/s eta 0:00:00
Requirement already satisfied: gensim<4.0.0 in
/usr/local/lib/python3.8/dist-packages (from pycaret) (3.6.0)
Collecting mlxtend>=0.17.0
  Downloading mlxtend-0.21.0-py2.py3-none-any.whl (1.3 MB)
                              1.3/1.3 MB
27.4 MB/s eta 0:00:00
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.8/dist-packages (from pycaret) (3.2.2)
Collecting scikit-learn==0.23.2
  Downloading scikit_learn-0.23.2-cp38-cp38-manylinux1_x86_64.whl (6.8 MB)
                              6.8/6.8 MB
26.7 MB/s eta 0:00:00
Requirement already satisfied: ipywidgets in
/usr/local/lib/python3.8/dist-packages (from pycaret) (7.7.1)
Collecting scipy<=1.5.4
  Downloading scipy-1.5.4-cp38-cp38-manylinux1_x86_64.whl (25.8 MB)
                              25.8/25.8 MB
15.8 MB/s eta 0:00:00
Requirement already satisfied: wordcloud in /usr/local/lib/python3.8/dist-
packages (from pycaret) (1.8.2.2)
Requirement already satisfied: plotly>=4.4.1 in /usr/local/lib/python3.8/dist-
packages (from pycaret) (5.5.0)
Requirement already satisfied: yellowbrick>=1.0.1 in
/usr/local/lib/python3.8/dist-packages (from pycaret) (1.5)
Requirement already satisfied: IPython in /usr/local/lib/python3.8/dist-packages
(from pycaret) (7.9.0)
Collecting pyyaml<6.0.0
  Downloading PyYAML-5.4.1-cp38-cp38-manylinux1_x86_64.whl (662 kB)
                             662.4/662.4 KB
24.7 MB/s eta 0:00:00
Collecting Boruta
  Downloading Boruta-0.3-py3-none-any.whl (56 kB)
                              56.6/56.6 KB
598.1 kB/s eta 0:00:00
Collecting numba<0.55
  Downloading
numba-0.54.1-cp38-cp38-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (3.3 MB)
                              3.3/3.3 MB
31.5 MB/s eta 0:00:00
Requirement already satisfied: pandas in /usr/local/lib/python3.8/dist-
packages (from pycaret) (1.3.5)
Collecting kmodes>=0.10.1
  Downloading kmodes-0.12.2-py2.py3-none-any.whl (20 kB)
```

```
Collecting mlflow
  Downloading mlflow-2.1.1-py3-none-any.whl (16.7 MB)
                           16.7/16.7 MB
17.3 MB/s eta 0:00:00
Collecting pyLDAvis
  Downloading pyLDAvis-3.4.0-py3-none-any.whl (2.6 MB)
                           2.6/2.6 MB
19.0 MB/s eta 0:00:00
Collecting pyod
  Downloading pyod-1.0.7.tar.gz (147 kB)
                          147.7/147.7 KB
11.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting umap-learn
  Downloading umap-learn-0.5.3.tar.gz (88 kB)
                           88.2/88.2 KB
4.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting scikit-plot
  Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Requirement already satisfied: joblib in /usr/local/lib/python3.8/dist-packages
(from pycaret) (1.2.0)
Collecting spacy<2.4.0
  Downloading
spacy-2.3.9-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (5.0 MB)
                           5.0/5.0 MB
18.8 MB/s eta 0:00:00
Requirement already satisfied: seaborn in /usr/local/lib/python3.8/dist-
packages (from pycaret) (0.11.2)
Requirement already satisfied: textblob in /usr/local/lib/python3.8/dist-
packages (from pycaret) (0.15.3)
Collecting lightgbm>=2.3.1
  Downloading lightgbm-3.3.5-py3-none-manylinux1_x86_64.whl (2.0 MB)
                           2.0/2.0 MB
33.7 MB/s eta 0:00:00
Collecting imbalanced-learn==0.7.0
  Downloading imbalanced_learn-0.7.0-py3-none-any.whl (167 kB)
                          167.1/167.1

KB 8.9 MB/s eta 0:00:00
Collecting pandas-profiling>=2.8.0
  Downloading pandas_profiling-3.6.6-py2.py3-none-any.whl (324 kB)
                          324.4/324.4 KB
10.0 MB/s eta 0:00:00
Requirement already satisfied: nltk in /usr/local/lib/python3.8/dist-
packages (from pycaret) (3.7)
Requirement already satisfied: cufflinks>=0.17.0 in
/usr/local/lib/python3.8/dist-packages (from pycaret) (0.17.3)
```

```
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.8/dist-
packages (from imbalanced-learn==0.7.0->pycaret) (1.21.6)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.8/dist-packages (from scikit-learn==0.23.2->pycaret)
(3.1.0)
Requirement already satisfied: colorlover>=0.2.1 in
/usr/local/lib/python3.8/dist-packages (from cufflinks>=0.17.0->pycaret) (0.3.0)
Requirement already satisfied: setuptools>=34.4.1 in
/usr/local/lib/python3.8/dist-packages (from cufflinks>=0.17.0->pycaret)
(57.4.0)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.8/dist-
packages (from cufflinks>=0.17.0->pycaret) (1.15.0)
Requirement already satisfied: smart-open>=1.2.1 in
/usr/local/lib/python3.8/dist-packages (from gensim<4.0.0->pycaret) (6.3.0)
Requirement already satisfied: backcall in /usr/local/lib/python3.8/dist-
packages (from IPython->pycaret) (0.2.0)
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in
/usr/local/lib/python3.8/dist-packages (from IPython->pycaret) (2.0.10)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.8/dist-
packages (from IPython->pycaret) (0.7.5)
Requirement already satisfied: decorator in /usr/local/lib/python3.8/dist-
packages (from IPython->pycaret) (4.4.2)
Requirement already satisfied: pygments in /usr/local/lib/python3.8/dist-
packages (from IPython->pycaret) (2.6.1)
Collecting jedi>=0.10
  Downloading jedi-0.18.2-py2.py3-none-any.whl (1.6 MB)
                           1.6/1.6 MB
23.8 MB/s eta 0:00:00
Requirement already satisfied: pexpect in /usr/local/lib/python3.8/dist-
packages (from IPython->pycaret) (4.8.0)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.8/dist-
packages (from IPython->pycaret) (5.7.1)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in
/usr/local/lib/python3.8/dist-packages (from ipywidgets->pycaret) (3.0.5)
Requirement already satisfied: ipython-genutils~=0.2.0 in
/usr/local/lib/python3.8/dist-packages (from ipywidgets->pycaret) (0.2.0)
Requirement already satisfied: widgetsnbextension~=3.6.0 in
/usr/local/lib/python3.8/dist-packages (from ipywidgets->pycaret) (3.6.1)
Requirement already satisfied: ipykernel>=4.5.1 in
/usr/local/lib/python3.8/dist-packages (from ipywidgets->pycaret) (5.3.4)
Requirement already satisfied: wheel in /usr/local/lib/python3.8/dist-packages
(from lightgbm>=2.3.1->pycaret) (0.38.4)
Collecting mlxtend>=0.17.0
  Downloading mlxtend-0.20.0-py2.py3-none-any.whl (1.3 MB)
                           1.3/1.3 MB
37.0 MB/s eta 0:00:00
  Downloading mlxtend-0.19.0-py2.py3-none-any.whl (1.3 MB)
                           1.3/1.3 MB
```

```
23.6 MB/s eta 0:00:00
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib->pycaret) (3.0.9)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-
packages (from matplotlib->pycaret) (0.11.0)
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib->pycaret) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib->pycaret) (1.4.4)
Collecting llvmlite<0.38,>=0.37.0rc1
  Downloading llvmlite-0.37.0-cp38-cp38-manylinux2014_x86_64.whl (26.3 MB)
                           26.3/26.3 MB
13.8 MB/s eta 0:00:00
Collecting numpy>=1.13.3
  Downloading
numpy-1.20.3-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (15.4 MB)
                           15.4/15.4 MB
28.4 MB/s eta 0:00:00
Requirement already satisfied: pytz>=2017.3 in
/usr/local/lib/python3.8/dist-packages (from pandas->pycaret) (2022.7.1)
Collecting ydata-profiling
  Downloading ydata_profiling-4.0.0-py2.py3-none-any.whl (344 kB)
                          344.5/344.5 KB
21.9 MB/s eta 0:00:00
Requirement already satisfied: tenacity>=6.2.0 in
/usr/local/lib/python3.8/dist-packages (from plotly>=4.4.1->pycaret) (8.2.0)
Collecting thinc<7.5.0,>=7.4.1
  Downloading
thinc-7.4.6-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.1 MB)
                           1.1/1.1 MB
10.5 MB/s eta 0:00:00
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in
/usr/local/lib/python3.8/dist-packages (from spacy<2.4.0->pycaret) (0.10.1)
Collecting srsly<1.1.0,>=1.0.2
  Downloading
srsly-1.0.6-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (211 kB)
                          211.1/211.1

KB 3.1 MB/s eta 0:00:00
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in
/usr/local/lib/python3.8/dist-packages (from spacy<2.4.0->pycaret) (3.0.8)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in
/usr/local/lib/python3.8/dist-packages (from spacy<2.4.0->pycaret) (4.64.1)
Collecting catalogue<1.1.0,>=0.0.7
  Downloading catalogue-1.0.2-py2.py3-none-any.whl (16 kB)
Collecting plac<1.2.0,>=0.9.6
  Downloading plac-1.1.3-py2.py3-none-any.whl (20 kB)
Requirement already satisfied: blis<0.8.0,>=0.4.0 in
```

```
/usr/local/lib/python3.8/dist-packages (from spacy<2.4.0->pycaret) (0.7.9)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in
/usr/local/lib/python3.8/dist-packages (from spacy<2.4.0->pycaret) (2.25.1)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in
/usr/local/lib/python3.8/dist-packages (from spacy<2.4.0->pycaret) (2.0.7)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in
/usr/local/lib/python3.8/dist-packages (from spacy<2.4.0->pycaret) (1.0.9)
Collecting yellowbrick>=1.0.1
  Downloading yellowbrick-1.4-py3-none-any.whl (274 kB)
                           274.2/274.2

KB 4.9 MB/s eta 0:00:00
  Downloading yellowbrick-1.3.post1-py3-none-any.whl (271 kB)
                          271.4/271.4 KB
14.0 MB/s eta 0:00:00
Collecting numpy>=1.13.3
  Downloading numpy-1.19.5-cp38-cp38-manylinux2010_x86_64.whl (14.9 MB)
                           14.9/14.9 MB
20.3 MB/s eta 0:00:00
Collecting docker<7,>=4.0.0
  Downloading docker-6.0.1-py3-none-any.whl (147 kB)
                           147.5/147.5

KB 5.9 MB/s eta 0:00:00
Requirement already satisfied: pyarrow<11,>=4.0.0 in
/usr/local/lib/python3.8/dist-packages (from mlflow->pycaret) (9.0.0)
Requirement already satisfied: Jinja2<4,>=2.11 in /usr/local/lib/python3.8/dist-
packages (from mlflow->pycaret) (2.11.3)
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.8/dist-
packages (from mlflow->pycaret) (7.1.2)
Requirement already satisfied: sqlparse<1,>=0.4.0 in
/usr/local/lib/python3.8/dist-packages (from mlflow->pycaret) (0.4.3)
Collecting gitpython<4,>=2.1.0
  Downloading GitPython-3.1.31-py3-none-any.whl (184 kB)
                           184.3/184.3

KB 5.8 MB/s eta 0:00:00
Collecting databricks-cli<1,>=0.8.7
  Downloading databricks-cli-0.17.4.tar.gz (82 kB)
                           82.3/82.3 KB
3.8 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting gunicorn<21
  Downloading gunicorn-20.1.0-py3-none-any.whl (79 kB)
                           79.5/79.5 KB
2.1 MB/s eta 0:00:00
Collecting querystring-parser<2
  Downloading querystring_parser-1.2.4-py2.py3-none-any.whl (7.9 kB)
Requirement already satisfied: entrypoints<1 in /usr/local/lib/python3.8/dist-
```

```
packages (from mlflow->pycaret) (0.4)
Requirement already satisfied: cloudpickle<3 in /usr/local/lib/python3.8/dist-
packages (from mlflow->pycaret) (2.2.1)
Collecting alembic<2
  Downloading alembic-1.9.4-py3-none-any.whl (210 kB)
                           210.5/210.5

KB 9.8 MB/s eta 0:00:00
Requirement already satisfied: protobuf<5,>=3.12.0 in
/usr/local/lib/python3.8/dist-packages (from mlflow->pycaret) (3.19.6)
Requirement already satisfied: Flask<3 in /usr/local/lib/python3.8/dist-packages
(from mlflow->pycaret) (1.1.4)
Collecting packaging<23
  Downloading packaging-22.0-py3-none-any.whl (42 kB)
                           42.6/42.6 KB
1.5 MB/s eta 0:00:00
Requirement already satisfied: sqlalchemy<2,>=1.4.0 in
/usr/local/lib/python3.8/dist-packages (from mlflow->pycaret) (1.4.46)
Collecting importlib-metadata!=4.7.0,<6,>=3.7.0
  Downloading importlib_metadata-5.2.0-py3-none-any.whl (21 kB)
Requirement already satisfied: markdown<4,>=3.3 in
/usr/local/lib/python3.8/dist-packages (from mlflow->pycaret) (3.4.1)
Collecting shap<1,>=0.40
  Downloading
shap-0.41.0-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (575 kB)
                           575.9/575.9

KB 5.3 MB/s eta 0:00:00
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.8/dist-packages (from nltk->pycaret) (2022.6.2)
Requirement already satisfied: numexpr in /usr/local/lib/python3.8/dist-packages
(from pyLDAvis->pycaret) (2.8.4)
Collecting pyLDAvis
  Downloading pyLDAvis-3.3.1.tar.gz (1.7 MB)
                           1.7/1.7 MB
34.0 MB/s eta 0:00:00
  Installing build dependencies … done
  Getting requirements to build wheel … done
  Installing backend dependencies … done
  Preparing metadata (pyproject.toml) … done
Requirement already satisfied: future in /usr/local/lib/python3.8/dist-packages
(from pyLDAvis->pycaret) (0.16.0)
  Downloading pyLDAvis-3.3.0.tar.gz (1.7 MB)
                           1.7/1.7 MB
70.1 MB/s eta 0:00:00
  Installing build dependencies … done
  Getting requirements to build wheel … done
  Installing backend dependencies … done
```

```
  Preparing metadata (pyproject.toml) … done
  Downloading pyLDAvis-3.2.2.tar.gz (1.7 MB)
                           1.7/1.7 MB
60.2 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Collecting funcy
  Downloading funcy-1.18-py2.py3-none-any.whl (33 kB)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.8/dist-
packages (from pyod->pycaret) (0.12.2)
Collecting pynndescent>=0.5
  Downloading pynndescent-0.5.8.tar.gz (1.1 MB)
                           1.1/1.1 MB
55.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
Requirement already satisfied: pillow in /usr/local/lib/python3.8/dist-packages
(from wordcloud->pycaret) (7.1.2)
Requirement already satisfied: importlib-resources in
/usr/local/lib/python3.8/dist-packages (from alembic<2->mlflow->pycaret)
(5.10.2)
Collecting Mako
  Downloading Mako-1.2.4-py3-none-any.whl (78 kB)
                           78.7/78.7 KB
8.3 MB/s eta 0:00:00
Collecting pyjwt>=1.7.0
  Downloading PyJWT-2.6.0-py3-none-any.whl (20 kB)
Requirement already satisfied: oauthlib>=3.1.0 in /usr/local/lib/python3.8/dist-
packages (from databricks-cli<1,>=0.8.7->mlflow->pycaret) (3.2.2)
Requirement already satisfied: tabulate>=0.7.7 in /usr/local/lib/python3.8/dist-
packages (from databricks-cli<1,>=0.8.7->mlflow->pycaret) (0.8.10)
Collecting urllib3>=1.26.0
  Downloading urllib3-1.26.14-py2.py3-none-any.whl (140 kB)
                          140.6/140.6 KB
14.5 MB/s eta 0:00:00
Collecting requests<3.0.0,>=2.13.0
  Downloading requests-2.28.2-py3-none-any.whl (62 kB)
                           62.8/62.8 KB
7.0 MB/s eta 0:00:00
Collecting websocket-client>=0.32.0
  Downloading websocket_client-1.5.1-py3-none-any.whl (55 kB)
                           55.9/55.9 KB
6.4 MB/s eta 0:00:00
Requirement already satisfied: Werkzeug<2.0,>=0.15 in
/usr/local/lib/python3.8/dist-packages (from Flask<3->mlflow->pycaret) (1.0.1)
Requirement already satisfied: itsdangerous<2.0,>=0.24 in
/usr/local/lib/python3.8/dist-packages (from Flask<3->mlflow->pycaret) (1.1.0)
Collecting gitdb<5,>=4.0.1
  Downloading gitdb-4.0.10-py3-none-any.whl (62 kB)
                           62.7/62.7 KB
```

```
6.5 MB/s eta 0:00:00
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-
packages (from importlib-metadata!=4.7.0,<6,>=3.7.0->mlflow->pycaret) (3.12.1)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.8/dist-
packages (from ipykernel>=4.5.1->ipywidgets->pycaret) (6.1.12)
Requirement already satisfied: tornado>=4.2 in /usr/local/lib/python3.8/dist-
packages (from ipykernel>=4.5.1->ipywidgets->pycaret) (6.0.4)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in
/usr/local/lib/python3.8/dist-packages (from jedi>=0.10->IPython->pycaret)
(0.8.3)
Requirement already satisfied: MarkupSafe>=0.23 in
/usr/local/lib/python3.8/dist-packages (from Jinja2<4,>=2.11->mlflow->pycaret)
(2.0.1)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.8/dist-packages
(from prompt-toolkit<2.1.0,>=2.0.0->IPython->pycaret) (0.2.6)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.8/dist-
packages (from requests<3.0.0,>=2.13.0->spacy<2.4.0->pycaret) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<2.4.0->pycaret) (2022.12.7)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.8/dist-packages (from
requests<3.0.0,>=2.13.0->spacy<2.4.0->pycaret) (2.1.1)
Collecting slicer==0.0.7
  Downloading slicer-0.0.7-py3-none-any.whl (14 kB)
Requirement already satisfied: greenlet!=0.4.17 in
/usr/local/lib/python3.8/dist-packages (from
sqlalchemy<2,>=1.4.0->mlflow->pycaret) (2.0.2)
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.8/dist-
packages (from widgetsnbextension~=3.6.0->ipywidgets->pycaret) (5.7.16)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.8/dist-
packages (from pexpect->IPython->pycaret) (0.7.0)
Requirement already satisfied: patsy>=0.5 in /usr/local/lib/python3.8/dist-
packages (from statsmodels->pyod->pycaret) (0.5.3)
Collecting typeguard<2.14,>=2.13.2
  Downloading typeguard-2.13.3-py3-none-any.whl (17 kB)
Requirement already satisfied: pydantic<1.11,>=1.8.1 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas-
profiling>=2.8.0->pycaret) (1.10.4)
Collecting visions[type_image_path]==0.7.5
  Downloading visions-0.7.5-py3-none-any.whl (102 kB)
                          102.7/102.7 KB
11.7 MB/s eta 0:00:00
Collecting multimethod<1.10,>=1.4
  Downloading multimethod-1.9.1-py3-none-any.whl (10 kB)
Collecting statsmodels
  Downloading
statsmodels-0.13.5-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (9.9
```

```
MB)
                           9.9/9.9 MB
82.5 MB/s eta 0:00:00
Collecting phik<0.13,>=0.11.1
  Downloading
phik-0.12.3-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (679 kB)
                          679.5/679.5 KB
48.2 MB/s eta 0:00:00
Collecting htmlmin==0.1.12
  Downloading htmlmin-0.1.12.tar.gz (19 kB)
  Preparing metadata (setup.py) … done
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.8/dist-
packages (from visions[type_image_path]==0.7.5->ydata-profiling->pandas-
profiling>=2.8.0->pycaret) (3.0)
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.8/dist-
packages (from visions[type_image_path]==0.7.5->ydata-profiling->pandas-
profiling>=2.8.0->pycaret) (22.2.0)
Collecting tangled-up-in-unicode>=0.0.4
  Downloading tangled_up_in_unicode-0.2.0-py3-none-any.whl (4.7 MB)
                           4.7/4.7 MB
77.4 MB/s eta 0:00:00
Collecting imagehash
  Downloading ImageHash-4.3.1-py2.py3-none-any.whl (296 kB)
                          296.5/296.5 KB
30.2 MB/s eta 0:00:00
Collecting smmap<6,>=3.0.1
  Downloading smmap-5.0.0-py3-none-any.whl (24 kB)
Requirement already satisfied: terminado>=0.8.1 in
/usr/local/lib/python3.8/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret) (0.13.3)
Requirement already satisfied: nbformat in /usr/local/lib/python3.8/dist-
packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(5.7.3)
Requirement already satisfied: nbconvert<6.0 in /usr/local/lib/python3.8/dist-
packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(5.6.1)
Requirement already satisfied: pyzmq>=17 in /usr/local/lib/python3.8/dist-
packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(23.2.1)
Requirement already satisfied: Send2Trash in /usr/local/lib/python3.8/dist-
packages (from notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(1.8.0)
Requirement already satisfied: prometheus-client in
/usr/local/lib/python3.8/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret) (0.16.0)
Requirement already satisfied: jupyter-core>=4.4.0 in
/usr/local/lib/python3.8/dist-packages (from
notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret) (5.2.0)
```

```
Requirement already satisfied: typing-extensions>=4.2.0 in
/usr/local/lib/python3.8/dist-packages (from pydantic<1.11,>=1.8.1->ydata-
profiling->pandas-profiling>=2.8.0->pycaret) (4.4.0)
Requirement already satisfied: platformdirs>=2.5 in
/usr/local/lib/python3.8/dist-packages (from jupyter-
core>=4.4.0->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(3.0.0)
Requirement already satisfied: pandocfilters>=1.4.1 in
/usr/local/lib/python3.8/dist-packages (from
nbconvert<6.0->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(1.5.0)
Requirement already satisfied: testpath in /usr/local/lib/python3.8/dist-
packages (from
nbconvert<6.0->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(0.6.0)
Requirement already satisfied: bleach in /usr/local/lib/python3.8/dist-packages
(from
nbconvert<6.0->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(6.0.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.8/dist-
packages (from
nbconvert<6.0->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(0.7.1)
Requirement already satisfied: mistune<2,>=0.8.1 in
/usr/local/lib/python3.8/dist-packages (from
nbconvert<6.0->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(0.8.4)
Requirement already satisfied: fastjsonschema in /usr/local/lib/python3.8/dist-
packages (from
nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(2.16.2)
Requirement already satisfied: jsonschema>=2.6 in /usr/local/lib/python3.8/dist-
packages (from
nbformat->notebook>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret)
(4.3.3)
Requirement already satisfied: PyWavelets in /usr/local/lib/python3.8/dist-
packages (from imagehash->visions[type_image_path]==0.7.5->ydata-
profiling->pandas-profiling>=2.8.0->pycaret) (1.4.1)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in
/usr/local/lib/python3.8/dist-packages (from jsonschema>=2.6->nbformat->notebook
>=4.4.1->widgetsnbextension~=3.6.0->ipywidgets->pycaret) (0.19.3)
Requirement already satisfied: webencodings in /usr/local/lib/python3.8/dist-
packages (from bleach->nbconvert<6.0->notebook>=4.4.1->widgetsnbextension~=3.6.0
->ipywidgets->pycaret) (0.5.1)
Building wheels for collected packages: pyLDAvis, pyod, umap-learn, databricks-
cli, pynndescent, htmlmin
  Building wheel for pyLDAvis (setup.py) … done
  Created wheel for pyLDAvis: filename=pyLDAvis-3.2.2-py2.py3-none-any.whl
```

```
size=135618
sha256=43b5da7293f3cb46c64618339b612d10dffe058fbcbfe94d7c8c95e137b072b2
  Stored in directory: /root/.cache/pip/wheels/2a/5b/b3/26b52781cdeea9c815e147cf
d4ac4a0a3472bce92142115670
  Building wheel for pyod (setup.py) … done
  Created wheel for pyod: filename=pyod-1.0.7-py3-none-any.whl size=181101
sha256=507b13d877e010a5686bd9b686d8af3a0379da1126c93ec2ff0faad0be331083
  Stored in directory: /root/.cache/pip/wheels/f7/e2/c1/1c7fd8b261e72411f6509afb
429c84532e40ddcd96074473f4
  Building wheel for umap-learn (setup.py) … done
  Created wheel for umap-learn: filename=umap_learn-0.5.3-py3-none-any.whl
size=82829
sha256=783141b6e10593ef89583ad52382300a496e24381d1d214d0d0dfab8727de42b
  Stored in directory: /root/.cache/pip/wheels/a9/3a/67/06a8950e053725912e6a8c42
c4a3a241410f6487b8402542ea
  Building wheel for databricks-cli (setup.py) … done
  Created wheel for databricks-cli: filename=databricks_cli-0.17.4-py3-none-
any.whl size=142894
sha256=8e9dd732a7b1189a53da673441cf41cac0308f04daba1161730dc9dc08de1f6d
  Stored in directory: /root/.cache/pip/wheels/48/7c/6e/4bf2c1748c7ecf994ca95159
1de81674ed6bf633e1e337d873
  Building wheel for pynndescent (setup.py) … done
  Created wheel for pynndescent: filename=pynndescent-0.5.8-py3-none-any.whl
size=55513
sha256=00c477b808ea7039f0562f134aec4ed68db27633edc9d7301e659557e0204a4d
  Stored in directory: /root/.cache/pip/wheels/1c/63/3a/29954bca1a27ba100ed8c279
73a78cb71b43dc67aed62e80c3
  Building wheel for htmlmin (setup.py) … done
  Created wheel for htmlmin: filename=htmlmin-0.1.12-py3-none-any.whl size=27098
sha256=816699f7e07f84d90e33332f078240ed171667e43015517b55a9b3ef618c2854
  Stored in directory: /root/.cache/pip/wheels/23/14/6e/4be5bfeeb027f4939a01764b
48edd5996acf574b0913fe5243
Successfully built pyLDAvis pyod umap-learn databricks-cli pynndescent htmlmin
Installing collected packages: plac, htmlmin, funcy, websocket-client, urllib3,
typeguard, tangled-up-in-unicode, srsly, smmap, slicer, querystring-parser,
pyyaml, pyjwt, packaging, numpy, multimethod, Mako, llvmlite, jedi, importlib-
metadata, gunicorn, catalogue, scipy, requests, numba, gitdb, alembic, visions,
thinc, statsmodels, scikit-learn, pyLDAvis, phik, imagehash, gitpython, docker,
databricks-cli, yellowbrick, spacy, shap, scikit-plot, pyod, pynndescent,
mlxtend, lightgbm, kmodes, imbalanced-learn, Boruta, ydata-profiling, umap-
learn, mlflow, pandas-profiling, pycaret
  Attempting uninstall: urllib3
    Found existing installation: urllib3 1.24.3
    Uninstalling urllib3-1.24.3:
      Successfully uninstalled urllib3-1.24.3
  Attempting uninstall: typeguard
    Found existing installation: typeguard 2.7.1
    Uninstalling typeguard-2.7.1:
```

```
    Successfully uninstalled typeguard-2.7.1
Attempting uninstall: srsly
  Found existing installation: srsly 2.4.5
  Uninstalling srsly-2.4.5:
    Successfully uninstalled srsly-2.4.5
Attempting uninstall: pyyaml
  Found existing installation: PyYAML 6.0
  Uninstalling PyYAML-6.0:
    Successfully uninstalled PyYAML-6.0
Attempting uninstall: packaging
  Found existing installation: packaging 23.0
  Uninstalling packaging-23.0:
    Successfully uninstalled packaging-23.0
Attempting uninstall: numpy
  Found existing installation: numpy 1.21.6
  Uninstalling numpy-1.21.6:
    Successfully uninstalled numpy-1.21.6
Attempting uninstall: llvmlite
  Found existing installation: llvmlite 0.39.1
  Uninstalling llvmlite-0.39.1:
    Successfully uninstalled llvmlite-0.39.1
Attempting uninstall: importlib-metadata
  Found existing installation: importlib-metadata 6.0.0
  Uninstalling importlib-metadata-6.0.0:
    Successfully uninstalled importlib-metadata-6.0.0
Attempting uninstall: catalogue
  Found existing installation: catalogue 2.0.8
  Uninstalling catalogue-2.0.8:
    Successfully uninstalled catalogue-2.0.8
Attempting uninstall: scipy
  Found existing installation: scipy 1.7.3
  Uninstalling scipy-1.7.3:
    Successfully uninstalled scipy-1.7.3
Attempting uninstall: requests
  Found existing installation: requests 2.25.1
  Uninstalling requests-2.25.1:
    Successfully uninstalled requests-2.25.1
Attempting uninstall: numba
  Found existing installation: numba 0.56.4
  Uninstalling numba-0.56.4:
    Successfully uninstalled numba-0.56.4
Attempting uninstall: thinc
  Found existing installation: thinc 8.1.7
  Uninstalling thinc-8.1.7:
    Successfully uninstalled thinc-8.1.7
Attempting uninstall: statsmodels
  Found existing installation: statsmodels 0.12.2
  Uninstalling statsmodels-0.12.2:
```

```
    Successfully uninstalled statsmodels-0.12.2
Attempting uninstall: scikit-learn
  Found existing installation: scikit-learn 1.0.2
  Uninstalling scikit-learn-1.0.2:
    Successfully uninstalled scikit-learn-1.0.2
Attempting uninstall: yellowbrick
  Found existing installation: yellowbrick 1.5
  Uninstalling yellowbrick-1.5:
    Successfully uninstalled yellowbrick-1.5
Attempting uninstall: spacy
  Found existing installation: spacy 3.4.4
  Uninstalling spacy-3.4.4:
    Successfully uninstalled spacy-3.4.4
Attempting uninstall: mlxtend
  Found existing installation: mlxtend 0.14.0
  Uninstalling mlxtend-0.14.0:
    Successfully uninstalled mlxtend-0.14.0
Attempting uninstall: lightgbm
  Found existing installation: lightgbm 2.2.3
  Uninstalling lightgbm-2.2.3:
    Successfully uninstalled lightgbm-2.2.3
Attempting uninstall: imbalanced-learn
  Found existing installation: imbalanced-learn 0.8.1
  Uninstalling imbalanced-learn-0.8.1:
    Successfully uninstalled imbalanced-learn-0.8.1
Attempting uninstall: pandas-profiling
  Found existing installation: pandas-profiling 1.4.1
  Uninstalling pandas-profiling-1.4.1:
    Successfully uninstalled pandas-profiling-1.4.1
```

```
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
xarray 2022.12.0 requires numpy>=1.20, but you have numpy 1.19.5 which is
incompatible.
xarray-einstats 0.5.1 requires numpy>=1.20, but you have numpy 1.19.5 which is
incompatible.
xarray-einstats 0.5.1 requires scipy>=1.6, but you have scipy 1.5.4 which is
incompatible.
tensorflow 2.11.0 requires numpy>=1.20, but you have numpy 1.19.5 which is
incompatible.
jaxlib 0.3.25+cuda11.cudnn805 requires numpy>=1.20, but you have numpy 1.19.5
which is incompatible.
jax 0.3.25 requires numpy>=1.20, but you have numpy 1.19.5 which is
incompatible.
en-core-web-sm 3.4.1 requires spacy<3.5.0,>=3.4.0, but you have spacy 2.3.9
which is incompatible.
confection 0.0.4 requires srsly<3.0.0,>=2.4.0, but you have srsly 1.0.6 which is
incompatible.
cmdstanpy 1.1.0 requires numpy>=1.21, but you have numpy 1.19.5 which is
incompatible.
```

```
Successfully installed Boruta-0.3 Mako-1.2.4 alembic-1.9.4 catalogue-1.0.2
databricks-cli-0.17.4 docker-6.0.1 funcy-1.18 gitdb-4.0.10 gitpython-3.1.31
gunicorn-20.1.0 htmlmin-0.1.12 imagehash-4.3.1 imbalanced-learn-0.7.0 importlib-
metadata-5.2.0 jedi-0.18.2 kmodes-0.12.2 lightgbm-3.3.5 llvmlite-0.37.0
mlflow-2.1.1 mlxtend-0.19.0 multimethod-1.9.1 numba-0.54.1 numpy-1.19.5
packaging-22.0 pandas-profiling-3.6.6 phik-0.12.3 plac-1.1.3 pyLDAvis-3.2.2
pycaret-2.3.10 pyjwt-2.6.0 pynndescent-0.5.8 pyod-1.0.7 pyyaml-5.4.1
querystring-parser-1.2.4 requests-2.28.2 scikit-learn-0.23.2 scikit-plot-0.3.7
scipy-1.5.4 shap-0.41.0 slicer-0.0.7 smmap-5.0.0 spacy-2.3.9 srsly-1.0.6
statsmodels-0.13.5 tangled-up-in-unicode-0.2.0 thinc-7.4.6 typeguard-2.13.3
umap-learn-0.5.3 urllib3-1.26.14 visions-0.7.5 websocket-client-1.5.1 ydata-
profiling-4.0.0 yellowbrick-1.3.post1
```

[ ]:
```
import pandas as pd
```

```
df_path = 'https://raw.githubusercontent.com/DeepCodeSec/ml1000-p1/
↪working_models/data/winequality-white.csv'
dataset = pd.read_csv(df_path,
                      sep=';') #the separater in the raw data is ;. need to␣
↪indicate so columns are found
dataset.head()
```

[ ]:

```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.0              0.27         0.36            20.7      0.045
1            6.3              0.30         0.34             1.6      0.049
2            8.1              0.28         0.40             6.9      0.050
3            7.2              0.23         0.32             8.5      0.058
4            7.2              0.23         0.32             8.5      0.058

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 45.0                 170.0   1.0010  3.00       0.45
1                 14.0                 132.0   0.9940  3.30       0.49
2                 30.0                  97.0   0.9951  3.26       0.44
3                 47.0                 186.0   0.9956  3.19       0.40
4                 47.0                 186.0   0.9956  3.19       0.40

   alcohol  quality
0      8.8        6
1      9.5        6
2     10.1        6
3      9.9        6
4      9.9        6
```

### 1.1.1 Recode quality to a binary label

**Original**: quality of wine rated from 0-10 with 10 as the best

Above shows that the minimum rating was a 3 and max is 9. The mean and median are both ~6.

According to the website below, a rating of 7+ is good wine. https://vineroutes.com/wine-rating-system/#:~:text=Wines%20rated%2089%20and%20above,outstanding%20for%20its%20particular%20type.

**New**: For the purpose of classification, we recode quality to a binary label: 1 = 'high quality' if the quality rating was 7 or above, and 0 = 'standard' where the quality rating was 6 or lower.

[ ]:

```
import numpy as np

#add binary classification label
dataset['new_quality'] = np.where(dataset['quality'] > 6,
                                  1,
                                  0)
dataset.head(100)
```

[ ]:
```
    fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0             7.0             0.270         0.36            20.7      0.045
1             6.3             0.300         0.34             1.6      0.049
2             8.1             0.280         0.40             6.9      0.050
3             7.2             0.230         0.32             8.5      0.058
4             7.2             0.230         0.32             8.5      0.058
..            …                 …           …               …         …
95            7.1             0.260         0.29            12.4      0.044
96            6.0             0.340         0.66            15.9      0.046
97            8.6             0.265         0.36             1.2      0.034
98            9.8             0.360         0.46            10.5      0.038
99            6.0             0.340         0.66            15.9      0.046

    free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                  45.0                 170.0   1.0010  3.00       0.45
1                  14.0                 132.0   0.9940  3.30       0.49
2                  30.0                  97.0   0.9951  3.26       0.44
3                  47.0                 186.0   0.9956  3.19       0.40
4                  47.0                 186.0   0.9956  3.19       0.40
..                  …                    …        …     …          …
95                 62.0                 240.0   0.9969  3.04       0.42
96                 26.0                 164.0   0.9979  3.14       0.50
97                 15.0                  80.0   0.9913  2.95       0.36
98                  4.0                  83.0   0.9956  2.89       0.30
99                 26.0                 164.0   0.9979  3.14       0.50

    alcohol  quality  new_quality
0       8.8        6            0
1       9.5        6            0
2      10.1        6            0
3       9.9        6            0
4       9.9        6            0
..      …        …            …
95      9.2        6            0
96      8.8        6            0
97     11.4        7            1
98     10.1        4            0
99      8.8        6            0

[100 rows x 13 columns]
```

[ ]:
```
#drop old quality column and rename new
dataset = dataset.drop(columns=['quality']) #drops old column
dataset = dataset.rename(columns={'new_quality':'quality'}) #renames back to␣
↪quality

dataset.head() #double check it did what we asked
```

[ ]:
```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.0              0.27         0.36            20.7      0.045
1            6.3              0.30         0.34             1.6      0.049
2            8.1              0.28         0.40             6.9      0.050
3            7.2              0.23         0.32             8.5      0.058
4            7.2              0.23         0.32             8.5      0.058

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 45.0                 170.0   1.0010  3.00       0.45
1                 14.0                 132.0   0.9940  3.30       0.49
2                 30.0                  97.0   0.9951  3.26       0.44
3                 47.0                 186.0   0.9956  3.19       0.40
4                 47.0                 186.0   0.9956  3.19       0.40

   alcohol  quality
0      8.8        0
1      9.5        0
2     10.1        0
3      9.9        0
4      9.9        0
```

## 1.2 Exploratory analysis report

The code below automatically creates an exploratory data analysis repor which is output as an html file in the local files. What follows are the highlights of this EDA report:

[ ]:
```
#Load libraries for exploratory analysis
!pip3 install pandas_profiling --upgrade
import pandas_profiling
from pandas_profiling import ProfileReport
import pandas as pd

pr = ProfileReport(dataset)

pr.to_file(output_file="EDA.html")
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: pandas_profiling in
/usr/local/lib/python3.8/dist-packages (3.6.6)
Requirement already satisfied: ydata-profiling in /usr/local/lib/python3.8/dist-
packages (from pandas_profiling) (4.0.0)
Requirement already satisfied: typeguard<2.14,>=2.13.2 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(2.13.3)
Requirement already satisfied: phik<0.13,>=0.11.1 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
```

```
(0.12.3)
Requirement already satisfied: matplotlib<3.7,>=3.2 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(3.2.2)
Requirement already satisfied: scipy<1.10,>=1.4.1 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(1.5.4)
Requirement already satisfied: pydantic<1.11,>=1.8.1 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(1.10.4)
Requirement already satisfied: visions[type_image_path]==0.7.5 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(0.7.5)
Requirement already satisfied: numpy<1.24,>=1.16.0 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(1.19.5)
Requirement already satisfied: htmlmin==0.1.12 in /usr/local/lib/python3.8/dist-
packages (from ydata-profiling->pandas_profiling) (0.1.12)
Requirement already satisfied: requests<2.29,>=2.24.0 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(2.28.2)
Requirement already satisfied: seaborn<0.13,>=0.10.1 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(0.11.2)
Requirement already satisfied: multimethod<1.10,>=1.4 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(1.9.1)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(5.4.1)
Requirement already satisfied: tqdm<4.65,>=4.48.2 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(4.64.1)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(2.11.3)
Requirement already satisfied: pandas!=1.4.0,<1.6,>1.1 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(1.3.5)
Requirement already satisfied: statsmodels<0.14,>=0.13.2 in
/usr/local/lib/python3.8/dist-packages (from ydata-profiling->pandas_profiling)
(0.13.5)
Requirement already satisfied: tangled-up-in-unicode>=0.0.4 in
/usr/local/lib/python3.8/dist-packages (from
visions[type_image_path]==0.7.5->ydata-profiling->pandas_profiling) (0.2.0)
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.8/dist-
packages (from visions[type_image_path]==0.7.5->ydata-
profiling->pandas_profiling) (22.2.0)
```

```
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.8/dist-
packages (from visions[type_image_path]==0.7.5->ydata-
profiling->pandas_profiling) (3.0)
Requirement already satisfied: imagehash in /usr/local/lib/python3.8/dist-
packages (from visions[type_image_path]==0.7.5->ydata-
profiling->pandas_profiling) (4.3.1)
Requirement already satisfied: Pillow in /usr/local/lib/python3.8/dist-packages
(from visions[type_image_path]==0.7.5->ydata-profiling->pandas_profiling)
(7.1.2)
Requirement already satisfied: MarkupSafe>=0.23 in
/usr/local/lib/python3.8/dist-packages (from jinja2<3.2,>=2.11.1->ydata-
profiling->pandas_profiling) (2.0.1)
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib<3.7,>=3.2->ydata-
profiling->pandas_profiling) (2.8.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-
packages (from matplotlib<3.7,>=3.2->ydata-profiling->pandas_profiling) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib<3.7,>=3.2->ydata-
profiling->pandas_profiling) (1.4.4)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib<3.7,>=3.2->ydata-
profiling->pandas_profiling) (3.0.9)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-
packages (from pandas!=1.4.0,<1.6,>1.1->ydata-profiling->pandas_profiling)
(2022.7.1)
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/python3.8/dist-
packages (from phik<0.13,>=0.11.1->ydata-profiling->pandas_profiling) (1.2.0)
Requirement already satisfied: typing-extensions>=4.2.0 in
/usr/local/lib/python3.8/dist-packages (from pydantic<1.11,>=1.8.1->ydata-
profiling->pandas_profiling) (4.4.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.8/dist-packages (from requests<2.29,>=2.24.0->ydata-
profiling->pandas_profiling) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.8/dist-
packages (from requests<2.29,>=2.24.0->ydata-profiling->pandas_profiling) (2.10)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/usr/local/lib/python3.8/dist-packages (from requests<2.29,>=2.24.0->ydata-
profiling->pandas_profiling) (1.26.14)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.8/dist-packages (from requests<2.29,>=2.24.0->ydata-
profiling->pandas_profiling) (2022.12.7)
Requirement already satisfied: patsy>=0.5.2 in /usr/local/lib/python3.8/dist-
packages (from statsmodels<0.14,>=0.13.2->ydata-profiling->pandas_profiling)
(0.5.3)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.8/dist-
packages (from statsmodels<0.14,>=0.13.2->ydata-profiling->pandas_profiling)
(22.0)
```

```
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages
(from patsy>=0.5.2->statsmodels<0.14,>=0.13.2->ydata-
profiling->pandas_profiling) (1.15.0)
Requirement already satisfied: PyWavelets in /usr/local/lib/python3.8/dist-
packages (from imagehash->visions[type_image_path]==0.7.5->ydata-
profiling->pandas_profiling) (1.4.1)
```

```
<ipython-input-5-37cfb9f18440>:3: DeprecationWarning: `import pandas_profiling`
is going to be deprecated by April 1st. Please use `import ydata_profiling`
instead.
  import pandas_profiling
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]
```

```
Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]
```

```
Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]
```

```
Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]
```

[ ]:
```
#basic structure of dataframe
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4898 entries, 0 to 4897
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         4898 non-null   float64
 1   volatile acidity      4898 non-null   float64
 2   citric acid           4898 non-null   float64
 3   residual sugar        4898 non-null   float64
 4   chlorides             4898 non-null   float64
 5   free sulfur dioxide   4898 non-null   float64
 6   total sulfur dioxide  4898 non-null   float64
 7   density               4898 non-null   float64
 8   pH                    4898 non-null   float64
 9   sulphates             4898 non-null   float64
 10  alcohol               4898 non-null   float64
 11  quality               4898 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 459.3 KB
```

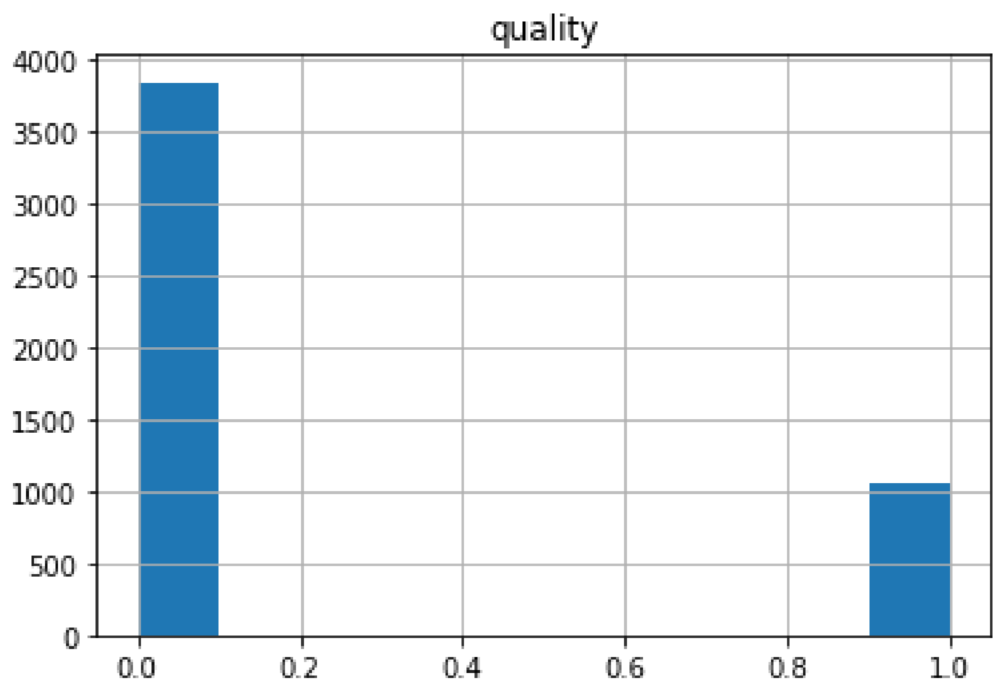
[ ]:
```
#check for missing values
dataset.isnull().sum()
```

[ ]:
```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

**Dataset has:** * 12 variables (11 numeric predictors and 1 categorical label that was recoded to binary 0 = standard, 1 = high quality) * 4898 observations * no missing values

[ ]:
```
#Distribution of target variable
dataset.hist(column='quality')
dataset[['quality']].describe()
```

[ ]:
```
           quality
count  4898.000000
mean      0.216415
std       0.411842
min       0.000000
25%       0.000000
50%       0.000000
75%       0.000000
max       1.000000
```




[ ]:
```
#Determine proportion of high quality wines in dataset
np.count_nonzero(dataset['quality']==1)/len(dataset['quality'])
```

[ ]: 0.21641486320947326

[ ]:
```
#Distribution of numeric predictors
dataset.describe()
```

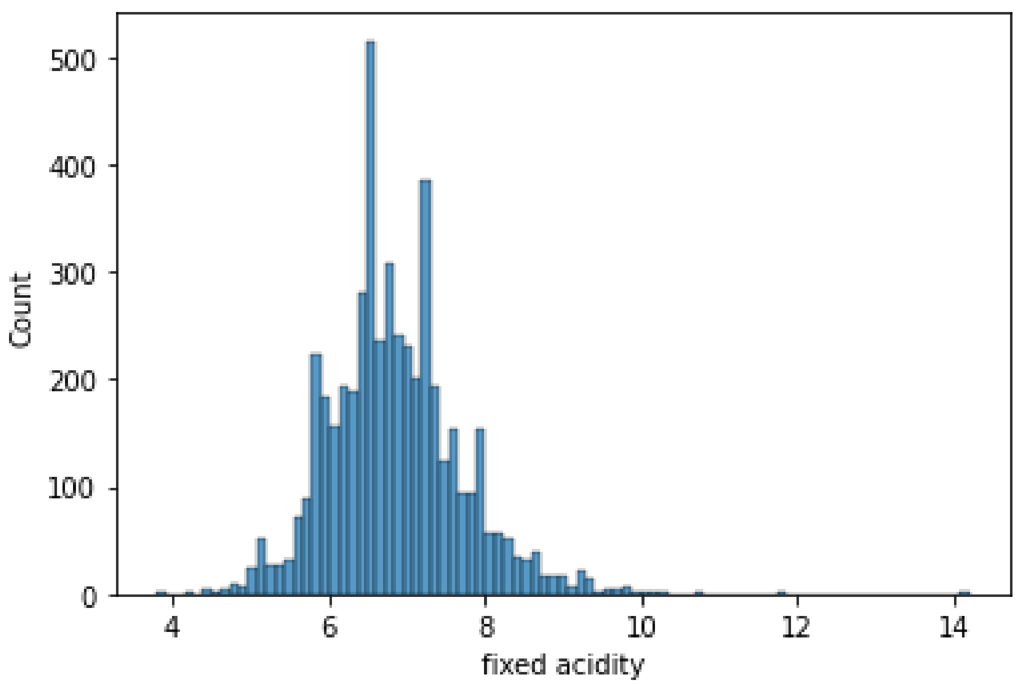
[ ]:

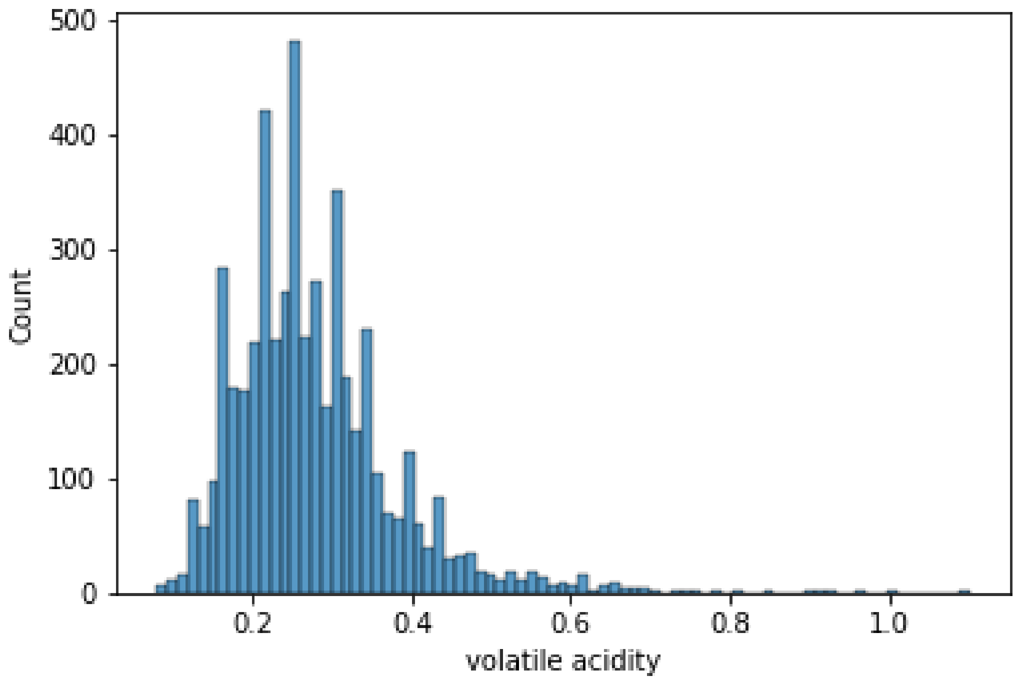
| | fixed acidity | volatile acidity | citric acid | residual sugar | \ |
|---|---|---|---|---|---|
| count | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 | |
| mean | 6.854788 | 0.278241 | 0.334192 | 6.391415 | |
| std | 0.843868 | 0.100795 | 0.121020 | 5.072058 | |
| min | 3.800000 | 0.080000 | 0.000000 | 0.600000 | |
| 25% | 6.300000 | 0.210000 | 0.270000 | 1.700000 | |
| 50% | 6.800000 | 0.260000 | 0.320000 | 5.200000 | |
| 75% | 7.300000 | 0.320000 | 0.390000 | 9.900000 | |
| max | 14.200000 | 1.100000 | 1.660000 | 65.800000 | |

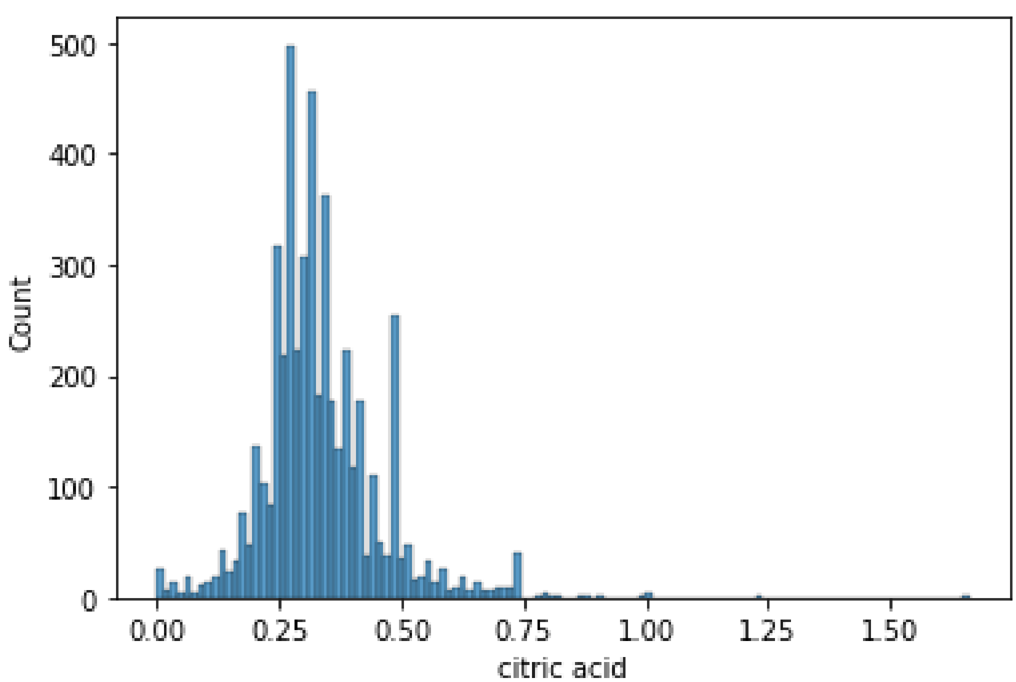
| | chlorides | free sulfur dioxide | total sulfur dioxide | density | \ |
|---|---|---|---|---|---|
| count | 4898.000000 | 4898.000000 | 4898.000000 | 4898.000000 | |
| mean | 0.045772 | 35.308085 | 138.360657 | 0.994027 | |
| std | 0.021848 | 17.007137 | 42.498065 | 0.002991 | |
| min | 0.009000 | 2.000000 | 9.000000 | 0.987110 | |
| 25% | 0.036000 | 23.000000 | 108.000000 | 0.991723 | |

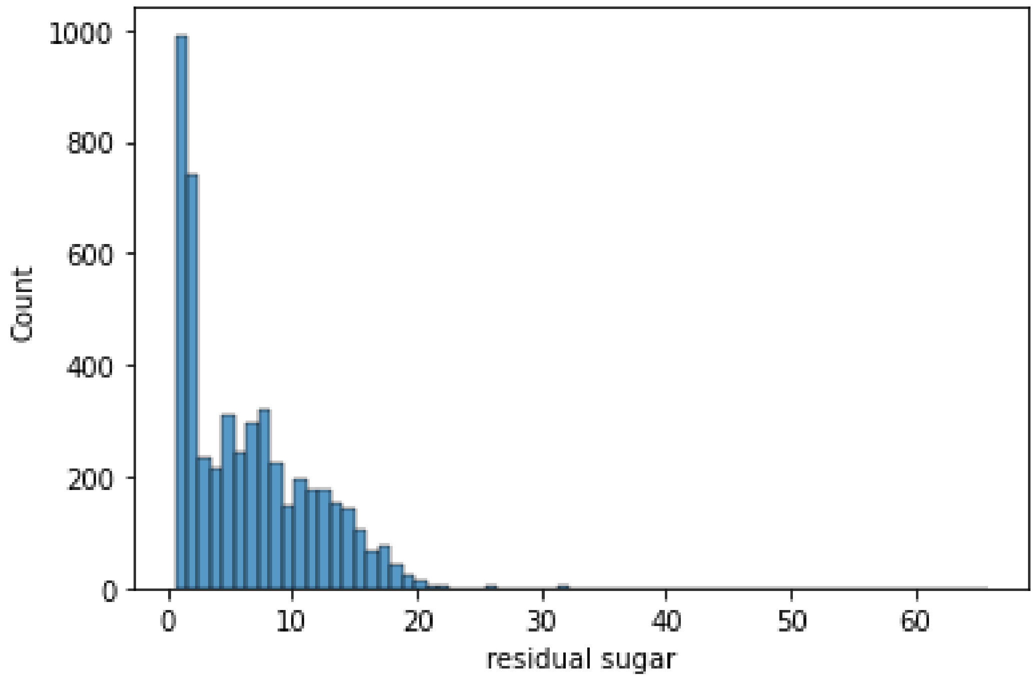
```
50%       0.043000            34.000000           134.000000     0.993740
75%       0.050000            46.000000           167.000000     0.996100
max       0.346000           289.000000           440.000000     1.038980

                pH    sulphates      alcohol      quality
count  4898.000000  4898.000000  4898.000000  4898.000000
mean      3.188267     0.489847    10.514267     0.216415
std       0.151001     0.114126     1.230621     0.411842
min       2.720000     0.220000     8.000000     0.000000
25%       3.090000     0.410000     9.500000     0.000000
50%       3.180000     0.470000    10.400000     0.000000
75%       3.280000     0.550000    11.400000     0.000000
max       3.820000     1.080000    14.200000     1.000000
```

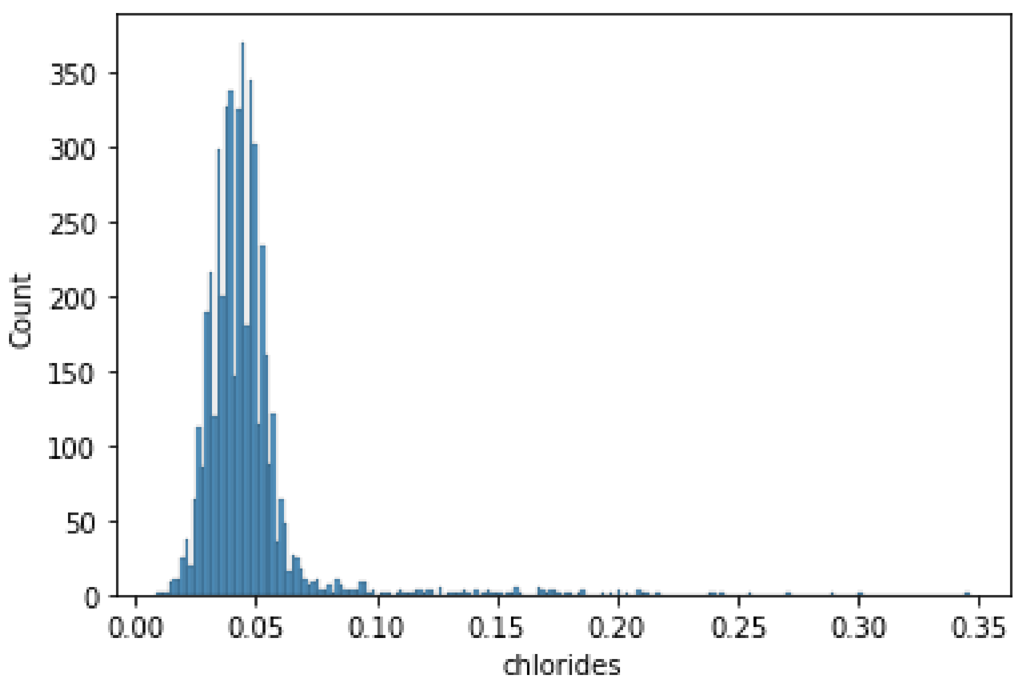
[ ]:
```
# Outlier analysis
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

cols = list(dataset.columns)

#Create boxplot for every numeric feature (cols 0-11) to show outliers
for col in cols[0:-1]:
  plt.figure()
  sns.histplot(dataset[col])
  plt.show()
```

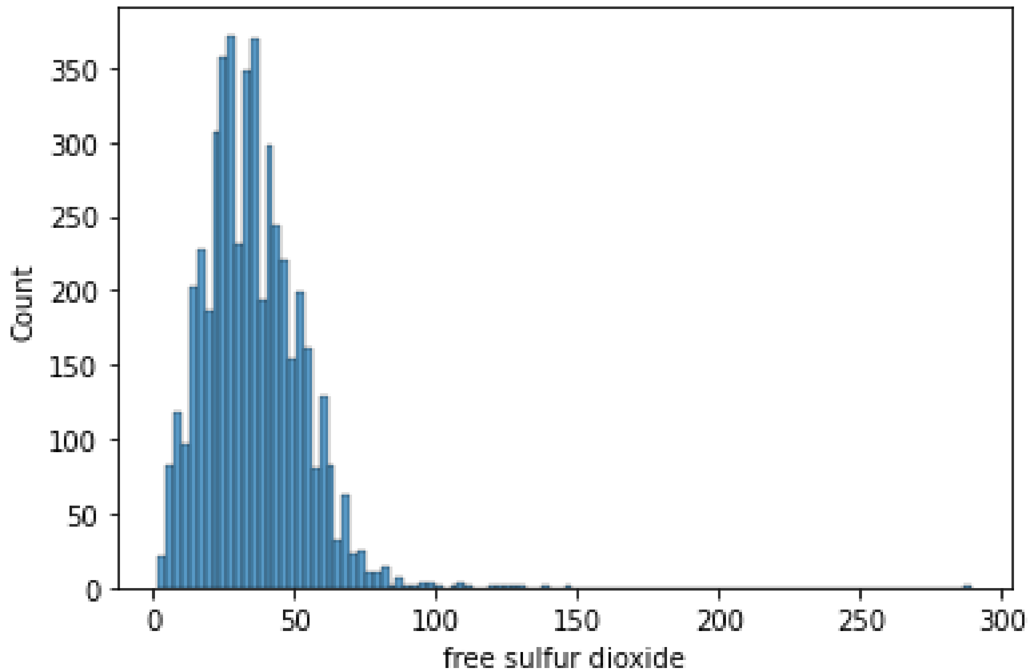



500
400
300
200
100
0
Count
0.2
0.4
0.6
0.8
1.0
volatile acidity

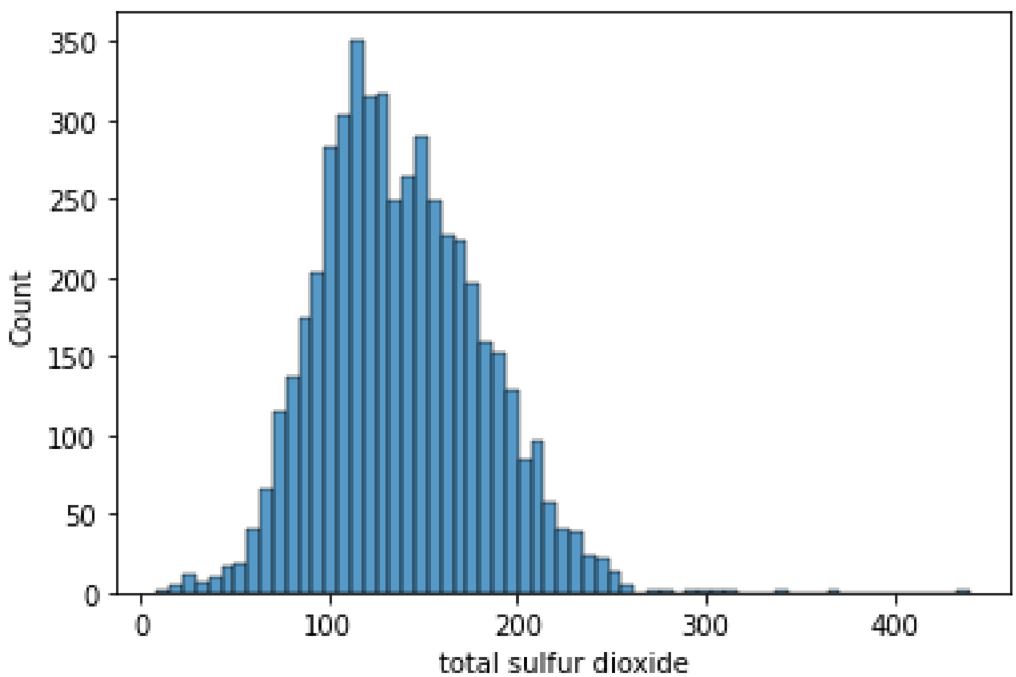

500
400
300
200
100
0
Count
0.00
0.25
0.50
0.75
1.00
1.25
1.50
citric acid


1000
800
600
400
200
0
Count
0
10
20
30
40
50
60
residual sugar

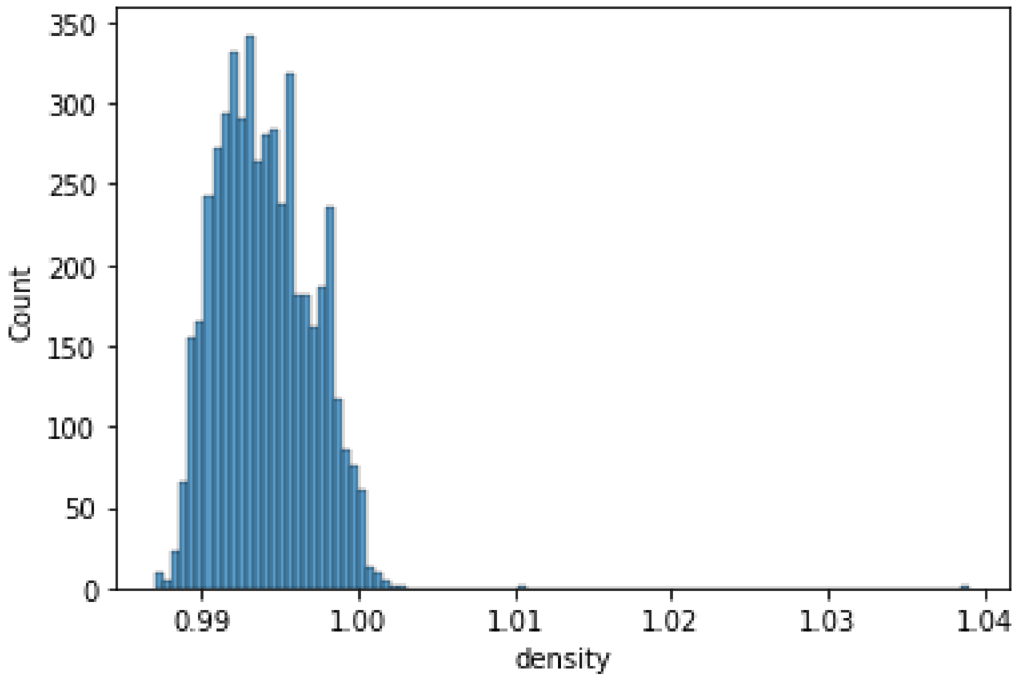

350
300
250
200
150
100
50
0
Count
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
chlorides

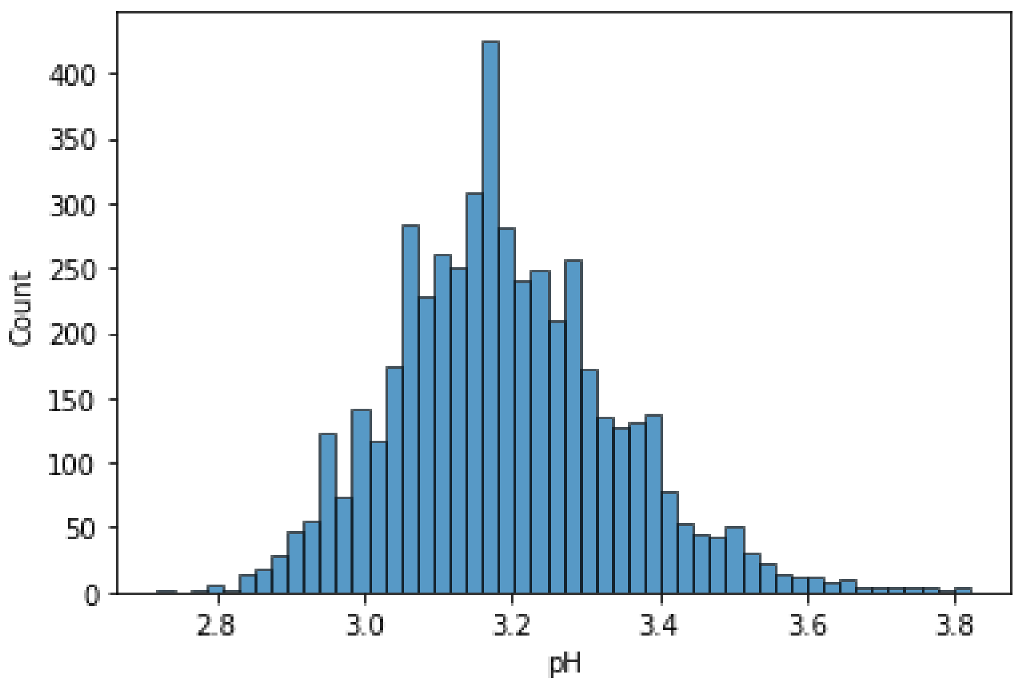


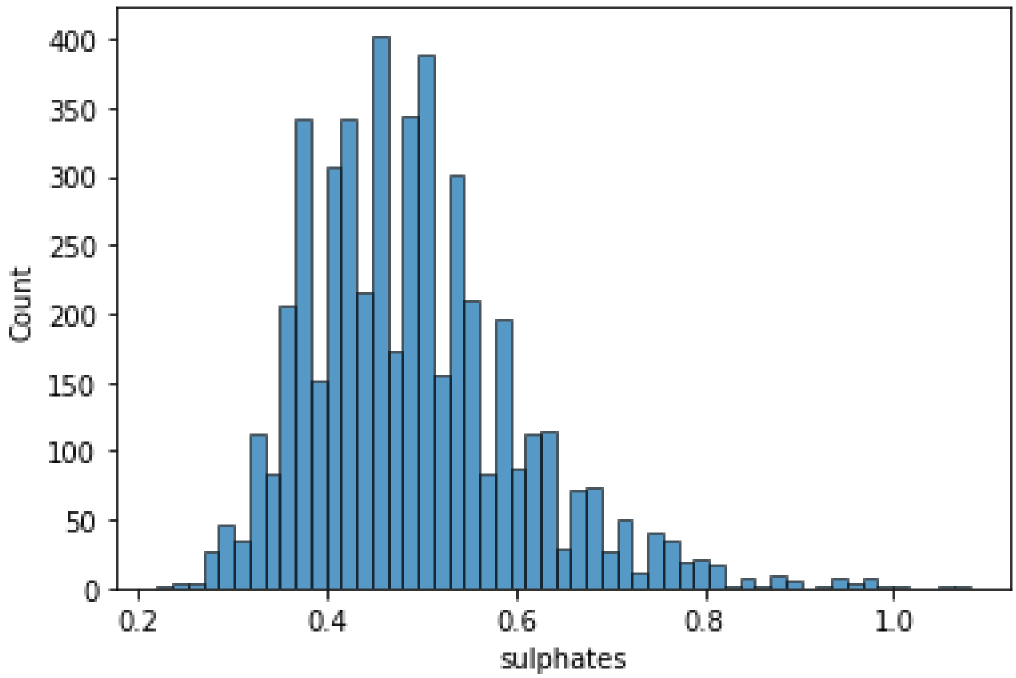
Count
free sulfur dioxide

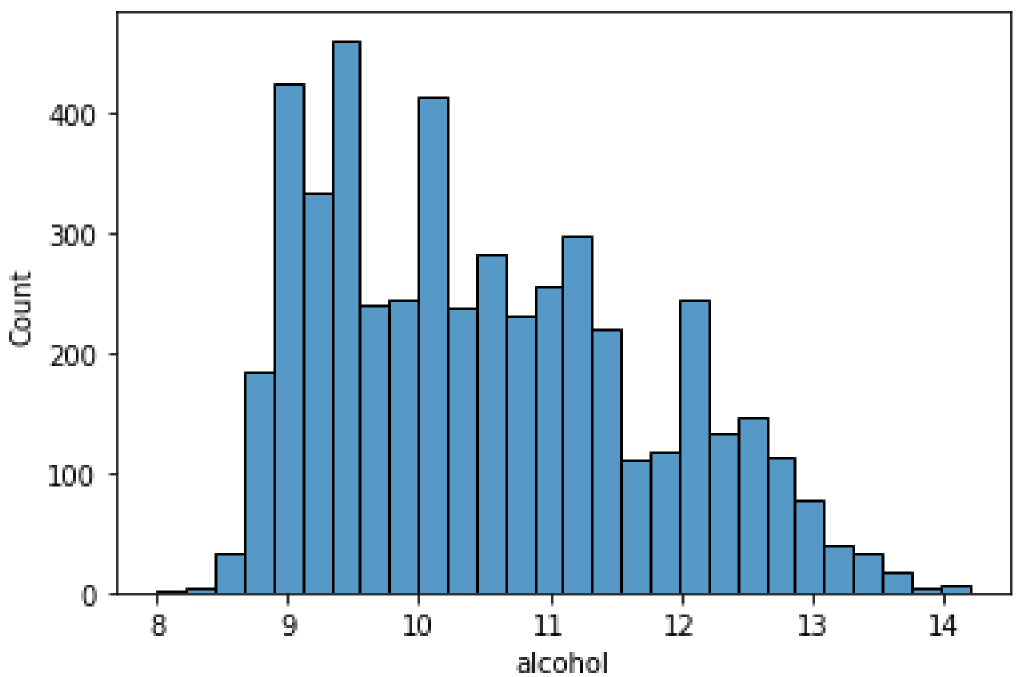



Count
total sulfur dioxide


350
300
250
200
150
100
50
0
Count
0.99
1.00
1.01
1.02
1.03
1.04
density


400
350
300
250
200
150
100
50
0
Count
2.8
3.0
3.2
3.4
3.6
3.8
pH

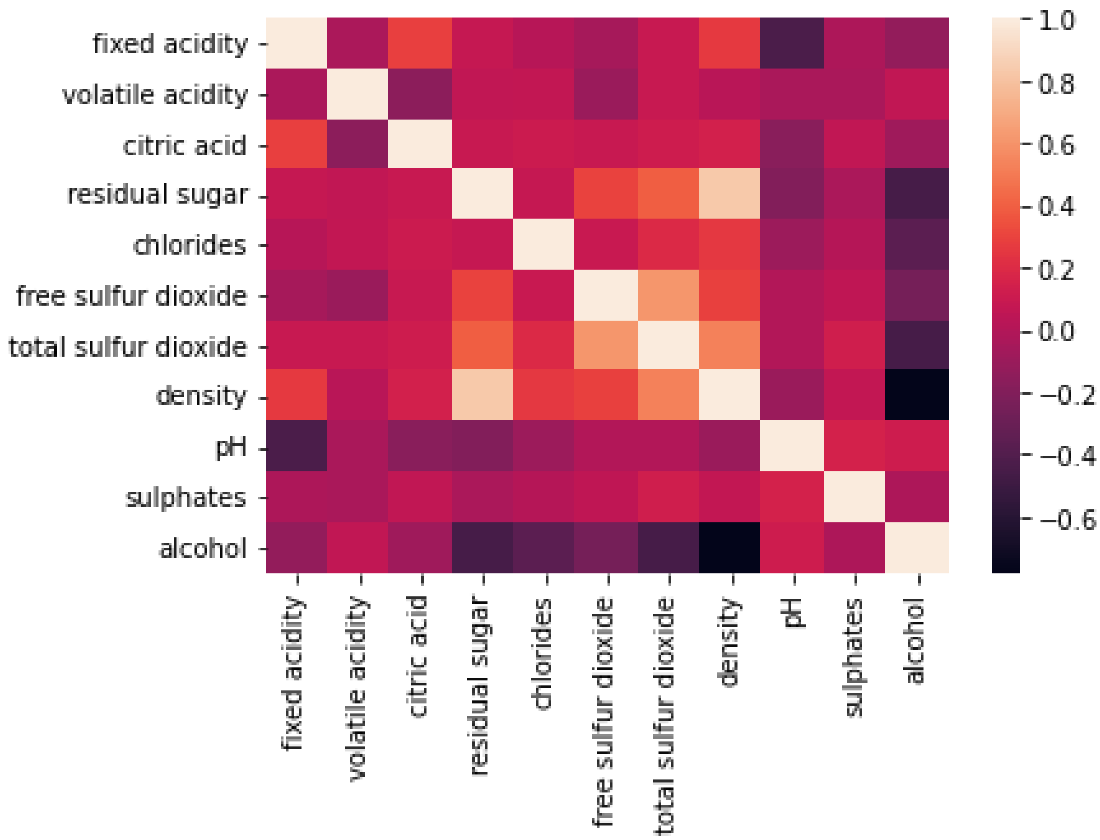



**Distributions**: * We have imbalanced label classes (~20% high quality and 80% standard) which indicates that we will need to think about undersampling and choose the appropriate performance metric to evaluate the trained models. * The numeric predictors are on vastly different scales which can skew the weigthing and importance of certain predictors. Therefore, we will need to scale the features before training the models. * Many predictors have a fairly normal distribution, but others such as alcohol, residual sugar, and citric acid are clearly non-normally distributed. We will apply transformations to make them look more normal in the experiment setup before training the models.

```
[ ]: #Correlation matrix for numeric predictor variables
     corr = dataset.iloc[:,:-1].corr()
     sns.heatmap(corr,
                 xticklabels=corr.columns.values,
                 yticklabels=corr.columns.values)
```

```
[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa51ac2e910>
```




**Correlations:** * Moderate to strong negative correlations between alcohol and density, and pH and fixed acidity. Strong positive correlation between density and residual sugar. We will address the multicollinearity in the experiment setup before training the models.

### 1.3 Split dataset

- 5% test set (unseen until after model is finalized)
- The remaining 95% will be split in the pycaret setup function

First we remove the 5% test set before any feature engineering to avoid data leakage. Test set is randomly shuffled.

After feature engineering, the remaining data will be split into training and validation sets that follow the same distribution of target labels (ie using stratified sampling).

[ ]:
```
# split data into 95% and 5%
data = dataset.sample(frac=0.95, random_state=786)
data_unseen = dataset.drop(data.index)
data.reset_index(inplace=True, drop=True)
data_unseen.reset_index(inplace=True, drop=True)
print('Data for Modeling: ' + str(data.shape))
print('Unseen Data For Predictions: ' + str(data_unseen.shape))
```

```
Data for Modeling: (4653, 12)
Unseen Data For Predictions: (245, 12)
```

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

[ ]:
```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            7.9              0.28         0.49             7.7      0.045
1            5.0              0.24         0.19             5.0      0.043
2            8.3              0.26         0.31             2.0      0.029
3            7.7              0.25         0.30             7.8      0.038
4            4.4              0.32         0.39             4.3      0.030

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 48.0                 195.0  0.99540  3.04       0.55
1                 17.0                 101.0  0.99438  3.67       0.57
2                 14.0                 141.0  0.99077  2.95       0.77
3                 67.0                 196.0  0.99555  3.10       0.50
4                 31.0                 127.0  0.98904  3.46       0.36

   alcohol  quality
0     11.0        0
1     10.0        0
2     12.2        0
3     10.1        0
4     12.8        1
```

[ ]:
```
data_unseen.head()
```

[ ]:
```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            8.1              0.28         0.40             6.9      0.050
1            8.6              0.23         0.40             4.2      0.035
2            6.6              0.16         0.40             1.5      0.044
3            7.4              0.34         0.42             1.1      0.033
4            6.0              0.19         0.26            12.4      0.048
```

```
   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 30.0                  97.0   0.9951  3.26       0.44
1                 17.0                 109.0   0.9947  3.14       0.53
2                 48.0                 143.0   0.9912  3.54       0.52
3                 17.0                 171.0   0.9917  3.12       0.53
4                 50.0                 147.0   0.9972  3.30       0.36

   alcohol  quality
0     10.1        0
1      9.7        0
2     12.4        1
3     11.3        0
4      8.9        0
```

### 1.4 Data Cleaning

Here we make our data cleaning decisions. As discussed above, we only have one categorical variable, the target label, which has already been one hot encoded. We do not need to do further categorical processing. Further, the data does not contain any missing values, so no imputation or row removal was needed. Exploratory data analysis revealed the numeric features have varying scales and several are non-normally distributed. We will address both in the experiment setup section. Finally, we need to check the data for outliers.

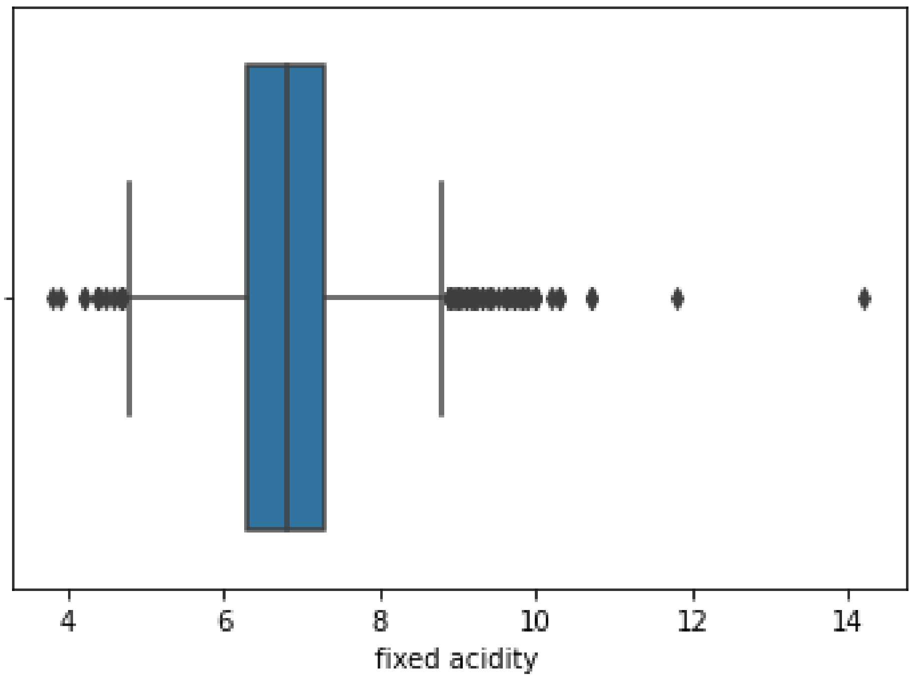
#### 1.4.1 Outlier analysis

Two options: 1. When lots of observations in dataset and only a few rows with outlier values (for any column), just remove rows containing outliers. Alternatively, we leave the feww outliers as is because they may be potentially informative.

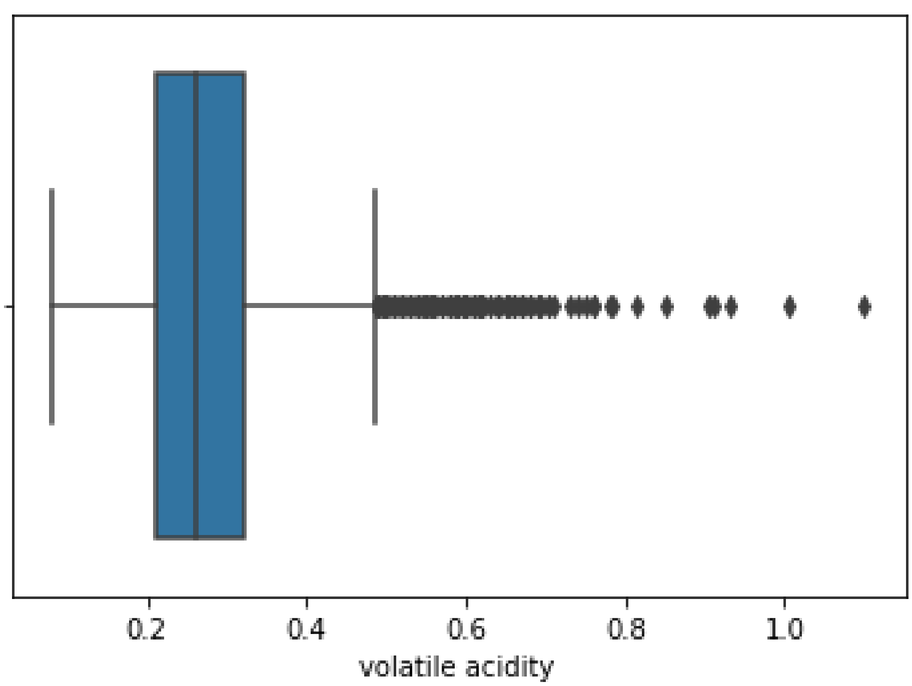
2. When fewer observations and more rows containing outlier values, cap the values at the 5th and 95th percentiles.

[ ]:
```
# Outlier analysis
cols = list(data.columns)

#Create boxplot for every numeric feature (cols 0-11) to show outliers
for col in cols[0:-1]:
  plt.figure()
  sns.boxplot(data[col])
  plt.show()
```

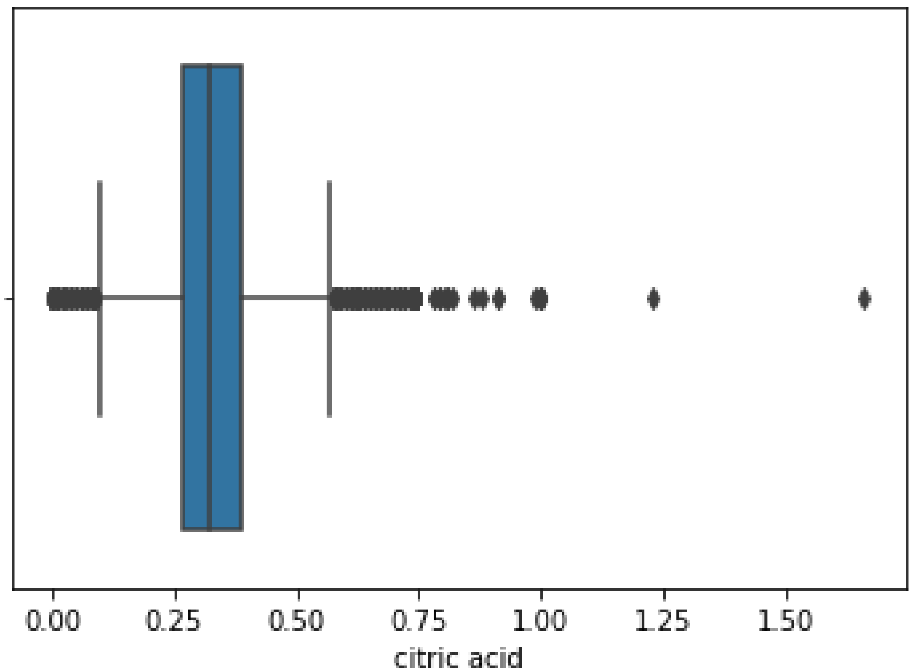
```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



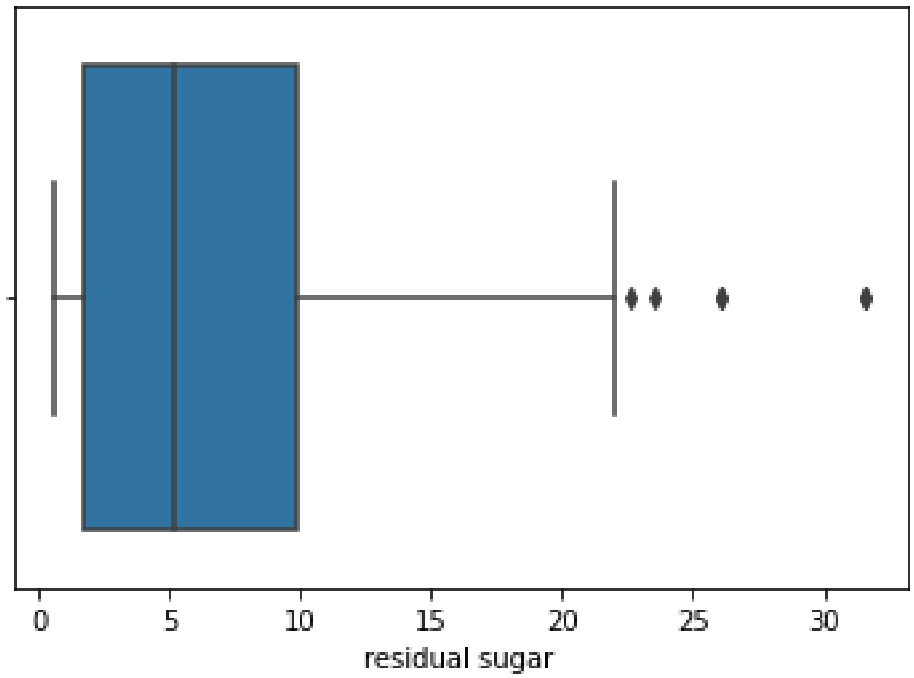

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



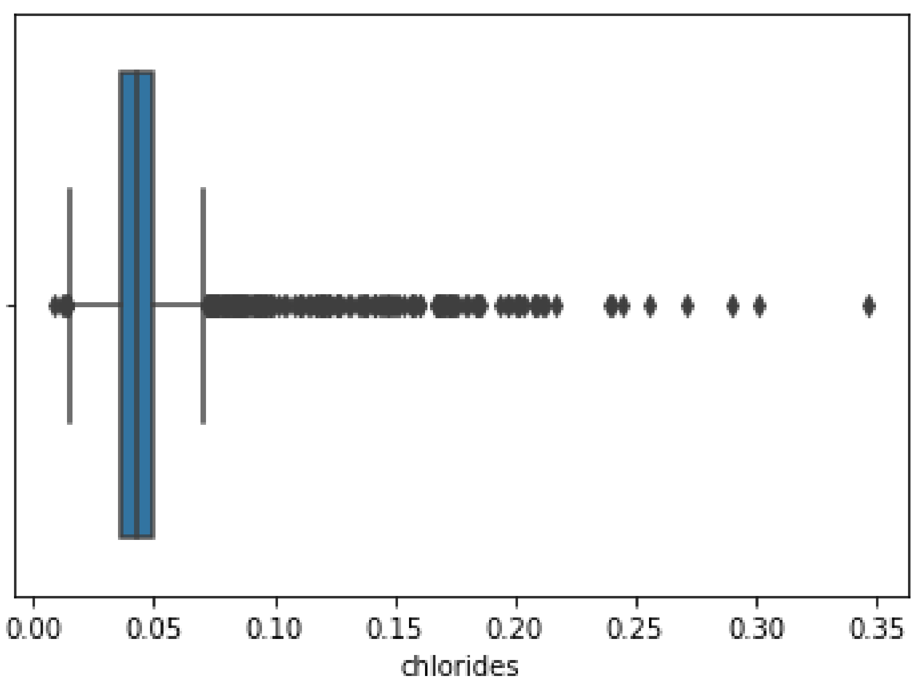
```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



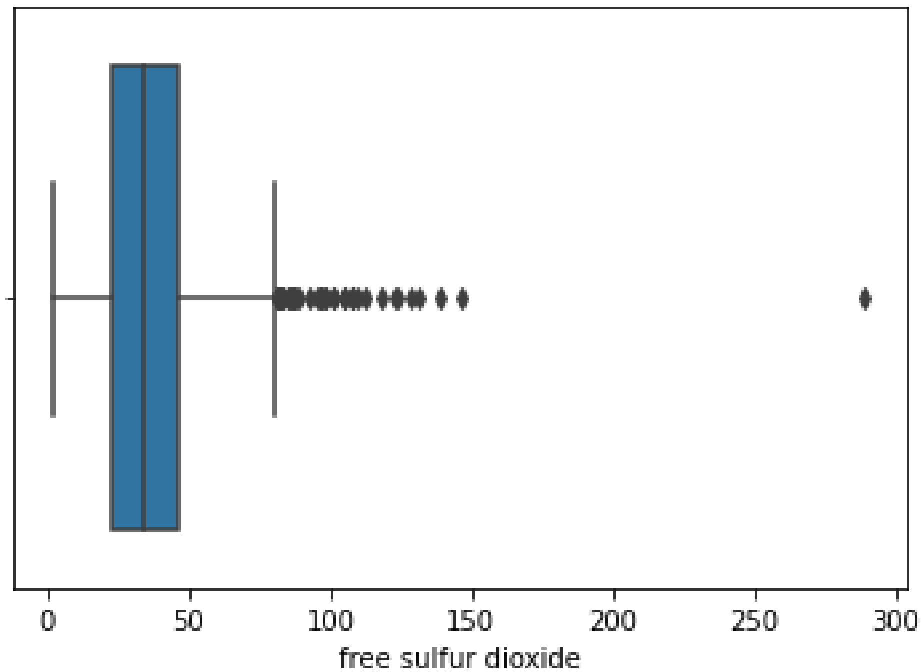

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



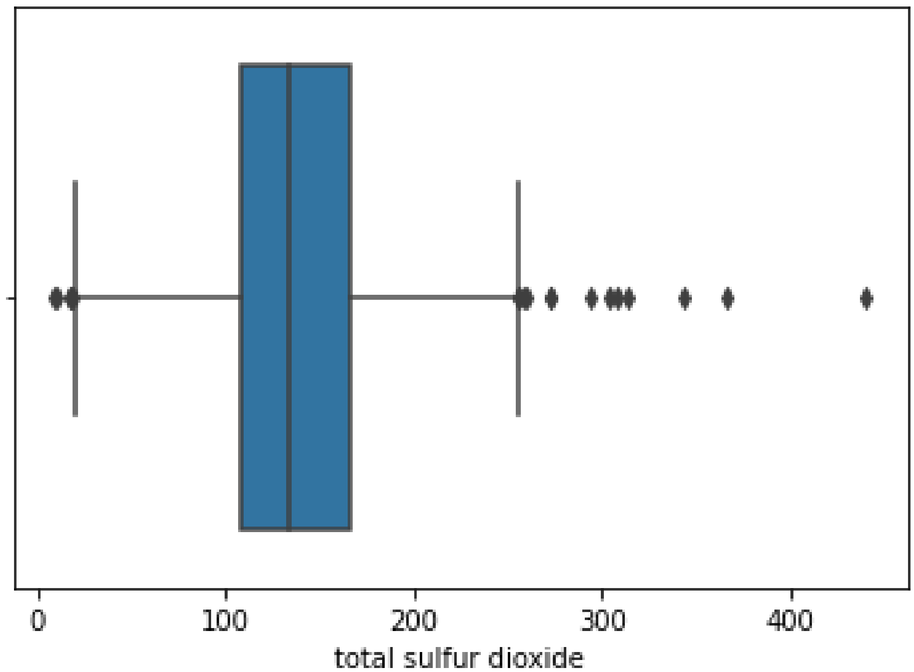

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



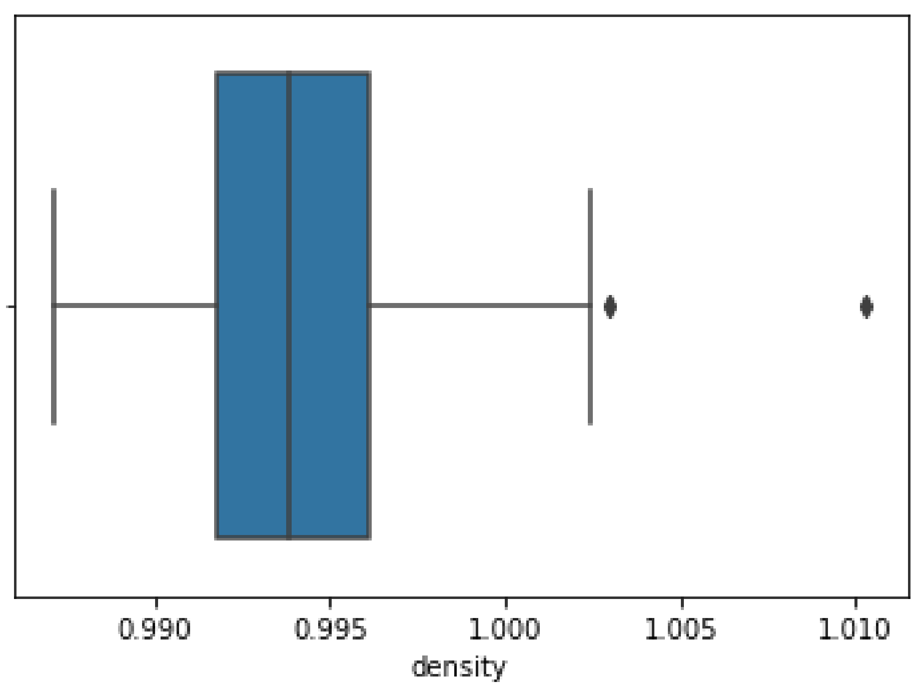
```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



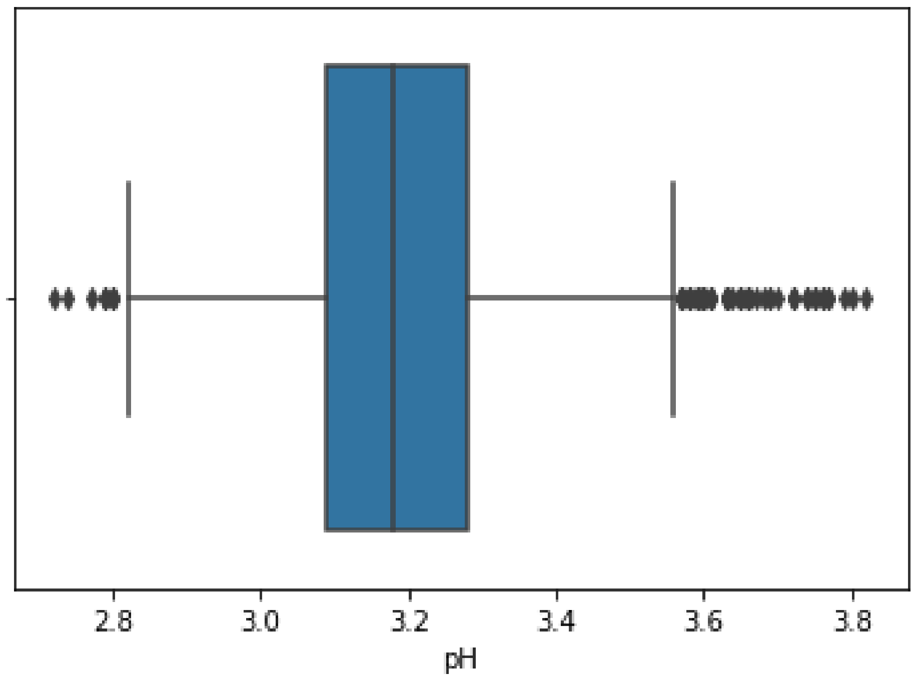

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



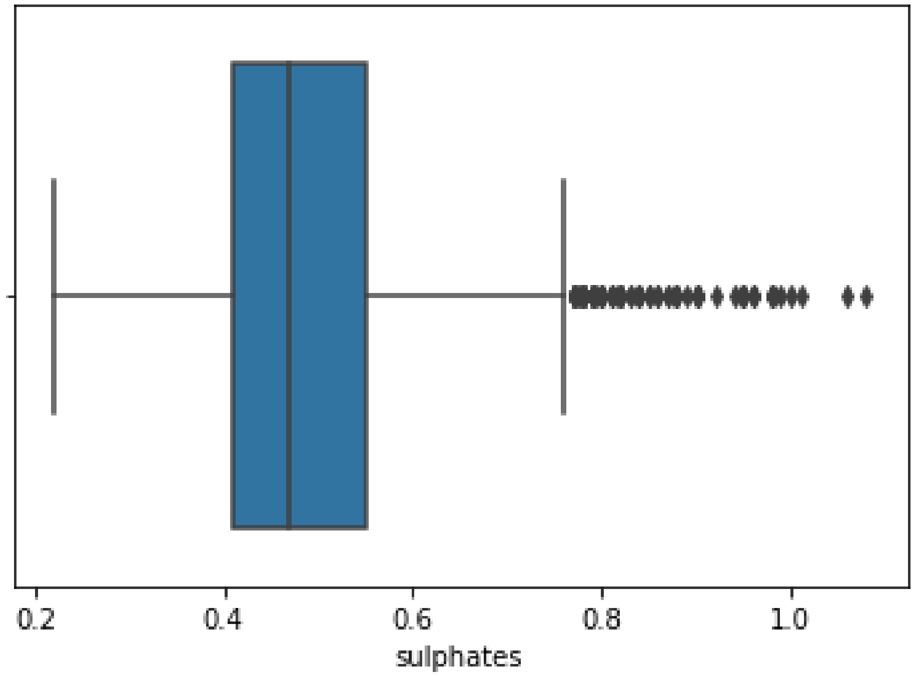

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



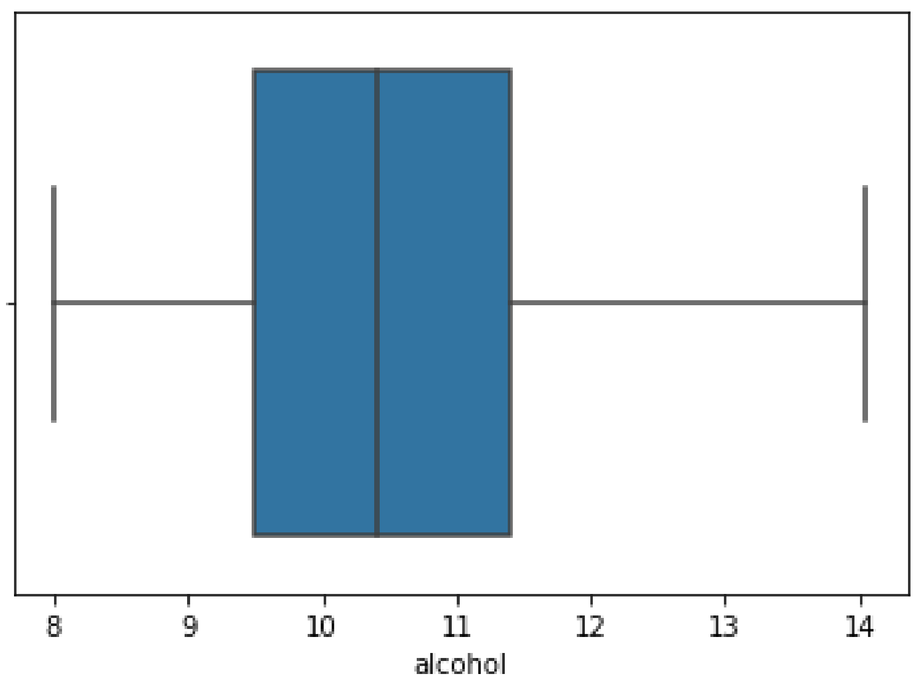
```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```




```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```




```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



The boxplots indicate that most of the predictors contain outlier values. Some of these predictors, like chlorides, contain a large number of outliers. Since there are so many outlier values, removing rows that contain outliers could leave us with a very small fraction of the original data. We opt to cap the values at the 5th and 95th percentiles to retain as much data as possible.

[ ]:
```
tmp = data #creating a temporary to avoid accidentally overwriting the original␣
↪(let's us compare and verify capping)
data_clean = data
```

[ ]:
```
data.describe()
```

[ ]:
```
       fixed acidity  volatile acidity  citric acid  residual sugar  \
count    4653.000000       4653.000000  4653.000000     4653.000000
mean        6.854406          0.277909     0.334030        6.405029
std         0.844952          0.100298     0.121024        5.008390
min         3.800000          0.080000     0.000000        0.600000
25%         6.300000          0.210000     0.270000        1.700000
50%         6.800000          0.260000     0.320000        5.200000
75%         7.300000          0.320000     0.390000        9.900000
max        14.200000          1.100000     1.660000       31.600000

         chlorides  free sulfur dioxide  total sulfur dioxide      density  \
count  4653.000000          4653.000000           4653.000000  4653.000000
mean      0.045796            35.343327            138.489792     0.994038
std       0.021997            17.025677             42.445410     0.002917
min       0.009000             2.000000              9.000000     0.987110
25%       0.036000            23.000000            108.000000     0.991750
50%       0.043000            34.000000            134.000000     0.993800
75%       0.050000            46.000000            167.000000     0.996120
max       0.346000           289.000000            440.000000     1.010300

                pH    sulphates      alcohol      quality
count  4653.000000  4653.000000  4653.000000  4653.000000
mean      3.187746     0.489695    10.504821     0.215345
std       0.149787     0.113718     1.227265     0.411106
min       2.720000     0.220000     8.000000     0.000000
25%       3.090000     0.410000     9.500000     0.000000
50%       3.180000     0.470000    10.400000     0.000000
75%       3.280000     0.550000    11.400000     0.000000
max       3.820000     1.080000    14.050000     1.000000
```

[ ]:
```
cols = list(data.columns)

#Create boxplot for every numeric feature (cols 0-11) to show outliers
for col in cols[0:-1]:
```

```
    upper_limit = tmp[col].mean() + 3*tmp[col].std() #~95th percentile
    lower_limit = tmp[col].mean() - 3*tmp[col].std() #~5th percentile

    data_clean[col] = np.where(tmp[col]> upper_limit, upper_limit, #if above␣
↪95th, set to upper
                      np.where(tmp[col]< lower_limit, lower_limit, #if below␣
↪5th, set to lower
                      tmp[col]))
```

[ ]:
```
#Capped distributions. Verify by checking max and min
data_clean.describe()
```

[ ]:

| | fixed acidity | volatile acidity | citric acid | residual sugar |
|---|---|---|---|---|
| count | 4653.000000 | 4653.000000 | 4653.000000 | 4653.000000 |
| mean | 6.850118 | 0.276365 | 0.332596 | 6.397731 |
| std | 0.825395 | 0.093928 | 0.114794 | 4.980920 |
| min | 4.319549 | 0.080000 | 0.000000 | 0.600000 |
| 25% | 6.300000 | 0.210000 | 0.270000 | 1.700000 |
| 50% | 6.800000 | 0.260000 | 0.320000 | 5.200000 |
| 75% | 7.300000 | 0.320000 | 0.390000 | 9.900000 |
| max | 9.389263 | 0.578803 | 0.697101 | 21.430199 |

| | chlorides | free sulfur dioxide | total sulfur dioxide | density |
|---|---|---|---|---|
| count | 4653.000000 | 4653.000000 | 4653.000000 | 4653.000000 |
| mean | 0.044657 | 35.177759 | 138.378840 | 0.994035 |
| std | 0.015199 | 16.129989 | 41.971721 | 0.002903 |
| min | 0.009000 | 2.000000 | 11.153562 | 0.987110 |
| 25% | 0.036000 | 23.000000 | 108.000000 | 0.991750 |
| 50% | 0.043000 | 34.000000 | 134.000000 | 0.993800 |
| 75% | 0.050000 | 46.000000 | 167.000000 | 0.996120 |
| max | 0.111788 | 86.420357 | 265.826021 | 1.002789 |

| | pH | sulphates | alcohol | quality |
|---|---|---|---|---|
| count | 4653.000000 | 4653.000000 | 4653.000000 | 4653.000000 |
| mean | 3.187297 | 0.488841 | 10.504821 | 0.215345 |
| std | 0.148243 | 0.110615 | 1.227265 | 0.411106 |
| min | 2.738384 | 0.220000 | 8.000000 | 0.000000 |
| 25% | 3.090000 | 0.410000 | 9.500000 | 0.000000 |
| 50% | 3.180000 | 0.470000 | 10.400000 | 0.000000 |
| 75% | 3.280000 | 0.550000 | 11.400000 | 0.000000 |
| max | 3.637107 | 0.830847 | 14.050000 | 1.000000 |

### 1.5 Training classifier models

Now that the data is clean, we can set up the classifier experiment pipeline. We address some final data cleaning issues in the experiment setup. For example, the values of the chlorides column ranges from 0.009 - 0.346 while residual sugar ranges from 0.6 - 65.8. This impacts performance on certain classifier algorithms, therefore we choose to z-score normalize the features to put them all on the same scale (-3 to +3). This is specified in the 'normalize=True' argument. Second, we use the 'transformation=True' arguement to transofrm the features into a more guassian (normal) distribution. Next, we address the multicollinearity in the data using the 'remove_multicollinearity' arguement. The threshold for multicollinearity was set to 0.7. Inter-correlated features that exceeded the 0.7 threshold were removed, and when two features are highly correlated with each other (for example, density and residual sugar) the feature that is least correlated with the target variable is removed. Finally, we address the imbalance in the target variable by employing the SMOTE algorithm which synthesises new examples from the minority class, in this case, high quality wines.

[ ]:
```
from pycaret.classification import *
```

[ ]:
```
exp_P1clf = setup(data = data_clean, #make sure to use cleaned data (outliers␣
↪capped)
                    target = 'quality', data_split_stratify = True, session_id =␣
↪123,
                    transformation=True, #applies the power transform to make␣
↪data more Gaussian-like
                    normalize=True, #transforms the numeric features by scaling␣
↪them to a given range (default is z-score)
                    remove_multicollinearity=True, #features with the␣
↪inter-correlations higher than the defined threshold are removed
                    multicollinearity_threshold = 0.7, #by default was 0.9
                    fix_imbalance=True #default method is SMOTE
                 )
```

| | Description | Value |
|---|---|---|
| 0 | session_id | 123 |
| 1 | Target | quality |
| 2 | Target Type | Binary |
| 3 | Label Encoded | None |
| 4 | Original Data | (4653, 12) |
| 5 | Missing Values | False |
| 6 | Numeric Features | 11 |
| 7 | Categorical Features | 0 |
| 8 | Ordinal Features | False |
| 9 | High Cardinality Features | False |
| 10 | High Cardinality Method | None |
| 11 | Transformed Train Set | (3257, 9) |
| 12 | Transformed Test Set | (1396, 9) |
| 13 | Shuffle Train-Test | True |
| 14 | Stratify Train-Test | True |
| 15 | Fold Generator | StratifiedKFold |
| 16 | Fold Number | 10 |
| 17 | CPU Jobs | -1 |
| 18 | Use GPU | False |
| 19 | Log Experiment | False |

```
20                     Experiment Name  clf-default-name
21                                 USI              b097
22                     Imputation Type            simple
23         Iterative Imputation Iteration            None
24                     Numeric Imputer              mean
25      Iterative Imputation Numeric Model          None
26                 Categorical Imputer          constant
27  Iterative Imputation Categorical Model          None
28          Unknown Categoricals Handling    least_frequent
29                           Normalize              True
30                    Normalize Method            zscore
31                      Transformation              True
32               Transformation Method       yeo-johnson
33                                 PCA             False
34                          PCA Method              None
35                      PCA Components              None
36                 Ignore Low Variance             False
37                 Combine Rare Levels             False
38                Rare Level Threshold              None
39                     Numeric Binning             False
40                     Remove Outliers             False
41                  Outliers Threshold              None
42             Remove Multicollinearity              True
43           Multicollinearity Threshold              0.7
44          Remove Perfect Collinearity              True
45                          Clustering             False
46                Clustering Iteration              None
47                 Polynomial Features             False
48                   Polynomial Degree              None
49                Trignometry Features             False
50                Polynomial Threshold              None
51                      Group Features             False
52                   Feature Selection             False
53             Feature Selection Method           classic
54          Features Selection Threshold             None
55                 Feature Interaction             False
56                       Feature Ratio             False
57               Interaction Threshold              None
58                       Fix Imbalance              True
59                Fix Imbalance Method             SMOTE

INFO:logs:create_model_container: 0
INFO:logs:master_model_container: 0
INFO:logs:display_container: 1
INFO:logs:Pipeline(memory=None,
         steps=[('dtypes',
                 DataTypes_Auto_infer(categorical_features=[],
```

```
                                display_types=True, features_todrop=[],
                                id_columns=[],
                                ml_usecase='classification',
                                numerical_features=[], target='quality',
                                time_features=[])),
                ('imputer',
                 Simple_Imputer(categorical_strategy='not_available',
                                fill_value_categorical=None,
                                fill_value_numerical=None,
                                numeric_stra…
                ('dummy', Dummify(target='quality')),
                ('fix_perfect', Remove_100(target='quality')),
                ('clean_names', Clean_Colum_Names()),
                ('feature_select', 'passthrough'),
                ('fix_multi',
                 Fix_multicollinearity(correlation_with_target_preference=None,
                                       correlation_with_target_threshold=0.0,
                                       target_variable='quality',
                                       threshold=0.7)),
                ('dfs', 'passthrough'), ('pca', 'passthrough')],
         verbose=False)
INFO:logs:setup() succesfully completed…
```

[ ]:
```
#Find best model
best_model = compare_models()
```

| | Model | Accuracy | AUC | Recall | Prec. | \ |
|---|---|---|---|---|---|---|
| et | Extra Trees Classifier | 0.8618 | 0.9110 | 0.6990 | 0.6731 | |
| rf | Random Forest Classifier | 0.8493 | 0.8993 | 0.6975 | 0.6375 | |
| lightgbm | Light Gradient Boosting Machine | 0.8394 | 0.8786 | 0.6177 | 0.6314 | |
| dt | Decision Tree Classifier | 0.8004 | 0.7466 | 0.6520 | 0.5297 | |
| dummy | Dummy Classifier | 0.7848 | 0.5000 | 0.0000 | 0.0000 | |
| gbc | Gradient Boosting Classifier | 0.7835 | 0.8475 | 0.7219 | 0.4983 | |
| ada | Ada Boost Classifier | 0.7562 | 0.8135 | 0.7162 | 0.4589 | |
| knn | K Neighbors Classifier | 0.7498 | 0.8262 | 0.7561 | 0.4521 | |
| lr | Logistic Regression | 0.7099 | 0.7818 | 0.7819 | 0.4088 | |
| ridge | Ridge Classifier | 0.7046 | 0.0000 | 0.7947 | 0.4051 | |
| lda | Linear Discriminant Analysis | 0.7046 | 0.7822 | 0.7947 | 0.4051 | |
| nb | Naive Bayes | 0.7016 | 0.7812 | 0.7732 | 0.4003 | |
| qda | Quadratic Discriminant Analysis | 0.6951 | 0.8050 | 0.7719 | 0.3941 | |
| svm | SVM - Linear Kernel | 0.6564 | 0.0000 | 0.7534 | 0.3605 | |

| | F1 | Kappa | MCC | TT (Sec) |
|---|---|---|---|---|
| et | 0.6829 | 0.5949 | 0.5972 | 0.455 |
| rf | 0.6642 | 0.5675 | 0.5699 | 1.158 |
| lightgbm | 0.6226 | 0.5209 | 0.5222 | 0.242 |
| dt | 0.5836 | 0.4544 | 0.4593 | 0.089 |
| dummy | 0.0000 | 0.0000 | 0.0000 | 0.026 |

```
gbc       0.5890  0.4489  0.4636     1.226
ada       0.5589  0.4018  0.4210     0.261
knn       0.5649  0.4048  0.4322     0.078
lr        0.5368  0.3544  0.3943     0.339
ridge     0.5365  0.3517  0.3951     0.042
lda       0.5365  0.3517  0.3951     0.033
nb        0.5271  0.3400  0.3799     0.043
qda       0.5214  0.3308  0.3716     0.029
svm       0.4860  0.2749  0.3185     0.069
```

```
INFO:logs:create_model_container: 14
INFO:logs:master_model_container: 14
INFO:logs:display_container: 2
INFO:logs:ExtraTreesClassifier(bootstrap=False, ccp_alpha=0.0,
class_weight=None,
                     criterion='gini', max_depth=None, max_features='auto',
                     max_leaf_nodes=None, max_samples=None,
                     min_impurity_decrease=0.0, min_impurity_split=None,
                     min_samples_leaf=1, min_samples_split=2,
                     min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=-1,
                     oob_score=False, random_state=123, verbose=0,
                     warm_start=False)
INFO:logs:compare_models() succesfully
completed…
```

The top three classifiers are extra trees, random forest, and light gradient boosting which show similar accuracy, auc and precision scores on the training data. Models preform differently on training data than they do on validation data, therefore we will tune and evaluate the top three models before seleting the best performer for the client.

Since our data is imbalanced, accuracy is not our most important performance metric. Since the client is most concerned with accurately predicting high quality wine (true positives), we will focus on the models' precision scores and use confusion matrices to evaluate performance.

[ ]:
```
# train a extra tree model
et = create_model('et')
```

| Fold | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| 0 | 0.8466 | 0.9086 | 0.5714 | 0.6667 | 0.6154 | 0.5203 | 0.5227 |
| 1 | 0.8650 | 0.9169 | 0.7571 | 0.6625 | 0.7067 | 0.6195 | 0.6219 |
| 2 | 0.8620 | 0.8993 | 0.6571 | 0.6866 | 0.6715 | 0.5842 | 0.5844 |
| 3 | 0.8620 | 0.9085 | 0.6286 | 0.6984 | 0.6617 | 0.5753 | 0.5765 |
| 4 | 0.8804 | 0.9313 | 0.7714 | 0.7013 | 0.7347 | 0.6577 | 0.6589 |
| 5 | 0.8497 | 0.8700 | 0.5714 | 0.6780 | 0.6202 | 0.5273 | 0.5303 |
| 6 | 0.8558 | 0.9076 | 0.7324 | 0.6500 | 0.6887 | 0.5954 | 0.5972 |
| 7 | 0.8954 | 0.9568 | 0.8714 | 0.7093 | 0.7821 | 0.7142 | 0.7207 |
| 8 | 0.8646 | 0.9213 | 0.7143 | 0.6757 | 0.6944 | 0.6076 | 0.6080 |

```
9        0.8369  0.8900  0.7143  0.6024  0.6536  0.5480  0.5514
Mean     0.8618  0.9110  0.6990  0.6731  0.6829  0.5949  0.5972
Std      0.0159  0.0222  0.0890  0.0294  0.0481  0.0564  0.0570
```

```
INFO:logs:create_model_container: 15
INFO:logs:master_model_container: 15
INFO:logs:display_container: 3
INFO:logs:ExtraTreesClassifier(bootstrap=False, ccp_alpha=0.0,
class_weight=None,
                     criterion='gini', max_depth=None, max_features='auto',
                     max_leaf_nodes=None, max_samples=None,
                     min_impurity_decrease=0.0, min_impurity_split=None,
                     min_samples_leaf=1, min_samples_split=2,
                     min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=-1,
                     oob_score=False, random_state=123, verbose=0,
                     warm_start=False)
INFO:logs:create_model() succesfully
completed…
```

[ ]:
```
# train a random forest model
rf = create_model('rf')
```

```
      Accuracy     AUC  Recall   Prec.      F1   Kappa     MCC
Fold
0       0.8405  0.8884  0.6857  0.6154  0.6486  0.5459  0.5472
1       0.8374  0.8989  0.7286  0.6000  0.6581  0.5527  0.5572
2       0.8405  0.8996  0.6143  0.6324  0.6232  0.5220  0.5221
3       0.8528  0.8967  0.6286  0.6667  0.6471  0.5541  0.5545
4       0.8528  0.9151  0.7429  0.6341  0.6842  0.5890  0.5921
5       0.8344  0.8573  0.5571  0.6290  0.5909  0.4875  0.4890
6       0.8558  0.9143  0.7465  0.6463  0.6928  0.5993  0.6019
7       0.8862  0.9471  0.8143  0.7037  0.7550  0.6813  0.6844
8       0.8523  0.9005  0.7000  0.6447  0.6712  0.5762  0.5770
9       0.8400  0.8755  0.7571  0.6023  0.6709  0.5670  0.5735
Mean    0.8493  0.8993  0.6975  0.6375  0.6642  0.5675  0.5699
Std     0.0143  0.0228  0.0736  0.0293  0.0414  0.0488  0.0494
```

```
INFO:logs:create_model_container: 16
INFO:logs:master_model_container: 16
INFO:logs:display_container: 4
INFO:logs:RandomForestClassifier(bootstrap=True, ccp_alpha=0.0,
class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
```

```
                        n_jobs=-1, oob_score=False, random_state=123, verbose=0,
                        warm_start=False)
INFO:logs:create_model() succesfully
completed…
```

[ ]:
```python
# train a lgb model
lgb = create_model('lightgbm')
```

| Fold | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| 0 | 0.8313 | 0.8808 | 0.6000 | 0.6087 | 0.6043 | 0.4971 | 0.4971 |
| 1 | 0.8405 | 0.8876 | 0.6143 | 0.6324 | 0.6232 | 0.5220 | 0.5221 |
| 2 | 0.8282 | 0.8688 | 0.5571 | 0.6094 | 0.5821 | 0.4743 | 0.4750 |
| 3 | 0.8497 | 0.8727 | 0.5429 | 0.6909 | 0.6080 | 0.5167 | 0.5224 |
| 4 | 0.8221 | 0.8928 | 0.6571 | 0.5750 | 0.6133 | 0.4985 | 0.5003 |
| 5 | 0.8344 | 0.8431 | 0.5571 | 0.6290 | 0.5909 | 0.4875 | 0.4890 |
| 6 | 0.8497 | 0.8886 | 0.6197 | 0.6667 | 0.6423 | 0.5474 | 0.5479 |
| 7 | 0.8862 | 0.9338 | 0.8000 | 0.7089 | 0.7517 | 0.6782 | 0.6803 |
| 8 | 0.8554 | 0.8758 | 0.6571 | 0.6667 | 0.6619 | 0.5699 | 0.5699 |
| 9 | 0.7969 | 0.8416 | 0.5714 | 0.5263 | 0.5479 | 0.4173 | 0.4179 |
| Mean | 0.8394 | 0.8786 | 0.6177 | 0.6314 | 0.6226 | 0.5209 | 0.5222 |
| Std | 0.0223 | 0.0249 | 0.0719 | 0.0522 | 0.0524 | 0.0655 | 0.0658 |

```
INFO:logs:create_model_container: 17
INFO:logs:master_model_container: 17
INFO:logs:display_container: 5
INFO:logs:LGBMClassifier(boosting_type='gbdt', class_weight=None,
colsample_bytree=1.0,
               importance_type='split', learning_rate=0.1, max_depth=-1,
               min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
               n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
               random_state=123, reg_alpha=0.0, reg_lambda=0.0, silent='warn',
               subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
INFO:logs:create_model() succesfully
completed…
```

### 1.6 Tuning the Models

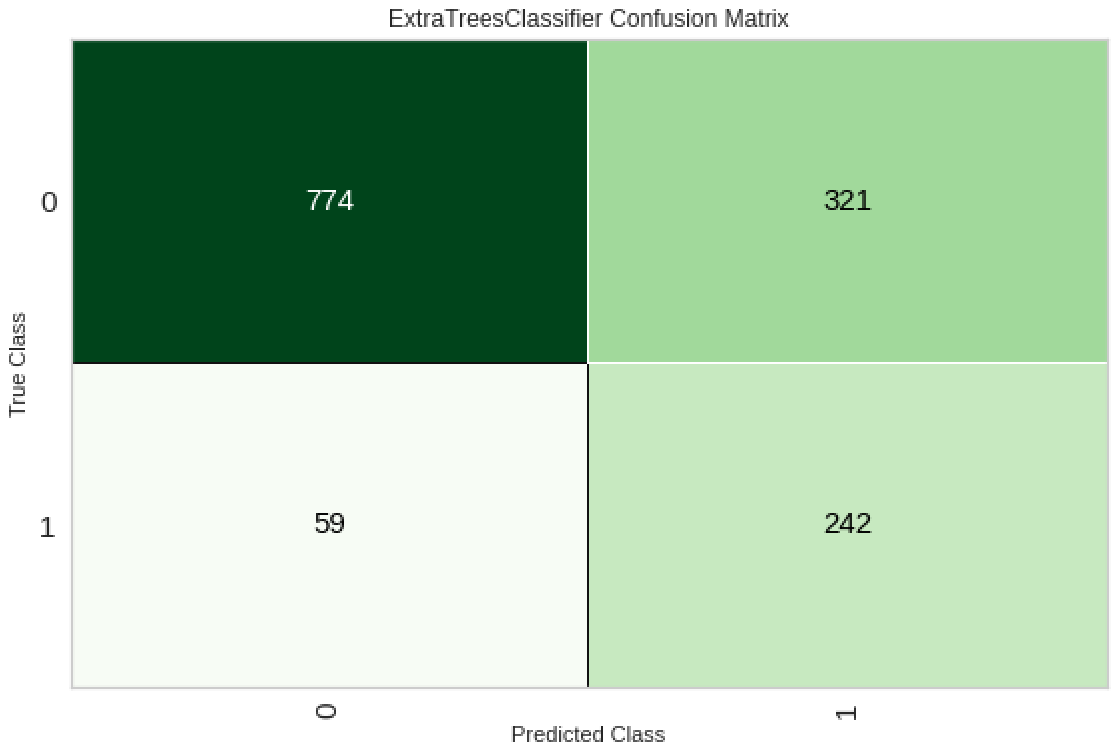
We use the tune_model function to find the optimal hyperparameters for the model. Our performance metric will be precision because the client is most concerned with the model's ability to identify true positives. That is, the LCBO wants to accurately price the high quality wines and maintain customer trust in their evaluations. If a high quality wine is misclassified as standard, the client loses profit. On the other hand, if a lower quality wine is overpriced it can erode the customer's trust in LCBO recommendations and negatively impact brand trust.

[ ]:
```python
# tune extra tree model
tuned_et = tune_model(et, optimize='Prec.')
```

```
      Accuracy     AUC  Recall   Prec.      F1   Kappa     MCC
Fold
0       0.7577  0.8511  0.8143  0.4634  0.5907  0.4364  0.4714
1       0.7546  0.8648  0.8714  0.4621  0.6040  0.4495  0.4970
2       0.7607  0.8509  0.7429  0.4643  0.5714  0.4175  0.4397
3       0.7546  0.8430  0.7857  0.4583  0.5789  0.4222  0.4528
4       0.7699  0.8337  0.7571  0.4775  0.5856  0.4375  0.4598
5       0.7270  0.8137  0.6714  0.4159  0.5137  0.3382  0.3569
6       0.7086  0.7877  0.6901  0.4016  0.5078  0.3207  0.3445
7       0.8092  0.8863  0.9143  0.5333  0.6737  0.5517  0.5917
8       0.7323  0.8411  0.7000  0.4261  0.5297  0.3578  0.3793
9       0.7262  0.8350  0.8429  0.4307  0.5700  0.3986  0.4470
Mean    0.7501  0.8407  0.7790  0.4533  0.5726  0.4130  0.4440
Std     0.0269  0.0255  0.0773  0.0355  0.0461  0.0626  0.0689
```

```
INFO:logs:create_model_container: 18
INFO:logs:master_model_container: 18
INFO:logs:display_container: 6
INFO:logs:ExtraTreesClassifier(bootstrap=False, ccp_alpha=0.0,
                     class_weight='balanced_subsample', criterion='gini',
                     max_depth=6, max_features=1.0, max_leaf_nodes=None,
                     max_samples=None, min_impurity_decrease=0,
                     min_impurity_split=None, min_samples_leaf=4,
                     min_samples_split=7, min_weight_fraction_leaf=0.0,
                     n_estimators=200, n_jobs=-1, oob_score=False,
                     random_state=123, verbose=0, warm_start=False)
INFO:logs:tune_model() succesfully
completed…
```

[ ]:
```
plot_model(tuned_et, plot = 'confusion_matrix')
```




```
INFO:logs:Visual Rendered Successfully
INFO:logs:plot_model() succesfully
completed…
```

Tuned extra trees model has %45 precision and 75% accuracy. The confusion matrix shows a higher rate of fase positives than true positives. This is potentially harmful to the client's brand trust. Conversly, there are few cases of false negatives, so the extra trees classifier reduces profit loss from underpricing high quality wines.

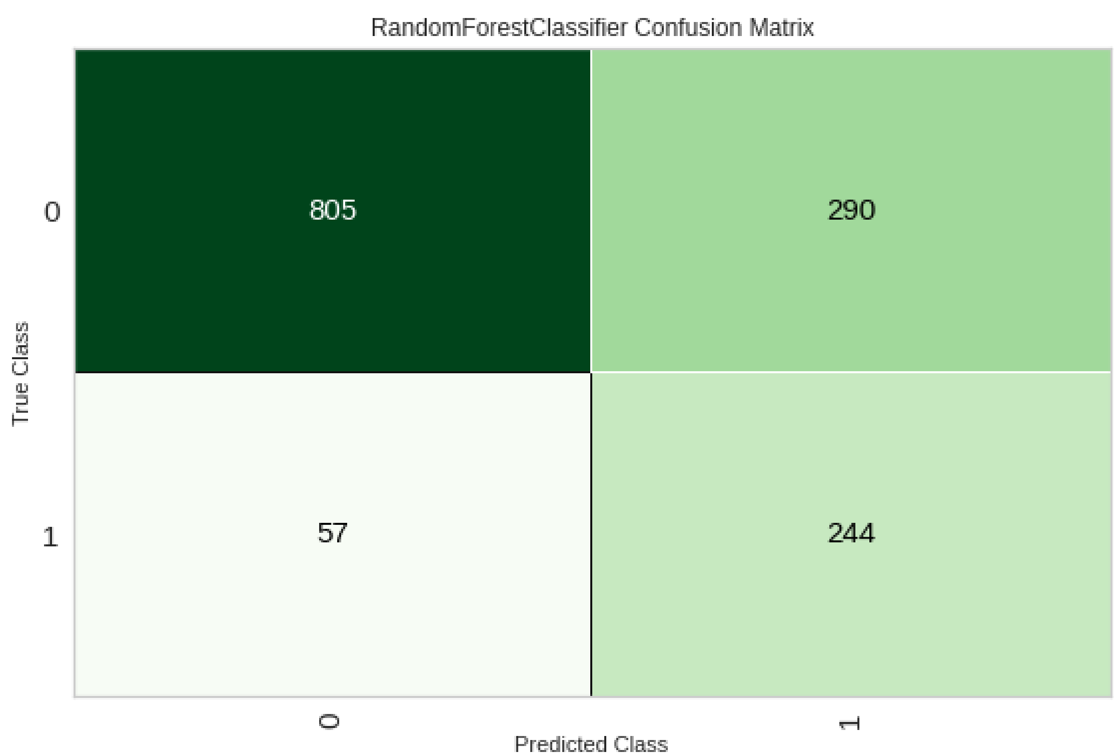
[ ]:
```
# tune random forest model
tuned_rf = tune_model(rf, optimize='Prec.')
```

| Fold | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| 0 | 0.7607 | 0.8488 | 0.8000 | 0.4667 | 0.5895 | 0.4367 | 0.4683 |
| 1 | 0.7423 | 0.8577 | 0.7714 | 0.4426 | 0.5625 | 0.3983 | 0.4292 |
| 2 | 0.7883 | 0.8522 | 0.7286 | 0.5050 | 0.5965 | 0.4594 | 0.4735 |
| 3 | 0.8006 | 0.8294 | 0.7286 | 0.5258 | 0.6108 | 0.4814 | 0.4930 |
| 4 | 0.7914 | 0.8459 | 0.7571 | 0.5096 | 0.6092 | 0.4742 | 0.4915 |
| 5 | 0.7270 | 0.7986 | 0.6143 | 0.4095 | 0.4914 | 0.3149 | 0.3270 |
| 6 | 0.7117 | 0.8130 | 0.6761 | 0.4034 | 0.5053 | 0.3197 | 0.3409 |
| 7 | 0.8185 | 0.8957 | 0.8857 | 0.5487 | 0.6776 | 0.5608 | 0.5919 |
| 8 | 0.7692 | 0.8389 | 0.6429 | 0.4737 | 0.5455 | 0.3955 | 0.4038 |
| 9 | 0.7538 | 0.8274 | 0.8143 | 0.4597 | 0.5876 | 0.4309 | 0.4668 |

```
Mean     0.7664  0.8407  0.7419  0.4745  0.5776  0.4272  0.4486
Std      0.0321  0.0253  0.0782  0.0457  0.0516  0.0708  0.0738
```

```
INFO:logs:create_model_container: 19
INFO:logs:master_model_container: 19
INFO:logs:display_container: 7
INFO:logs:RandomForestClassifier(bootstrap=False, ccp_alpha=0.0,
                       class_weight='balanced_subsample', criterion='gini',
                       max_depth=6, max_features='log2', max_leaf_nodes=None,
                       max_samples=None, min_impurity_decrease=0.001,
                       min_impurity_split=None, min_samples_leaf=6,
                       min_samples_split=9, min_weight_fraction_leaf=0.0,
                       n_estimators=190, n_jobs=-1, oob_score=False,
                       random_state=123, verbose=0, warm_start=False)
INFO:logs:tune_model() succesfully
completed…
```

[ ]:
```
plot_model(tuned_rf, plot = 'confusion_matrix')
```



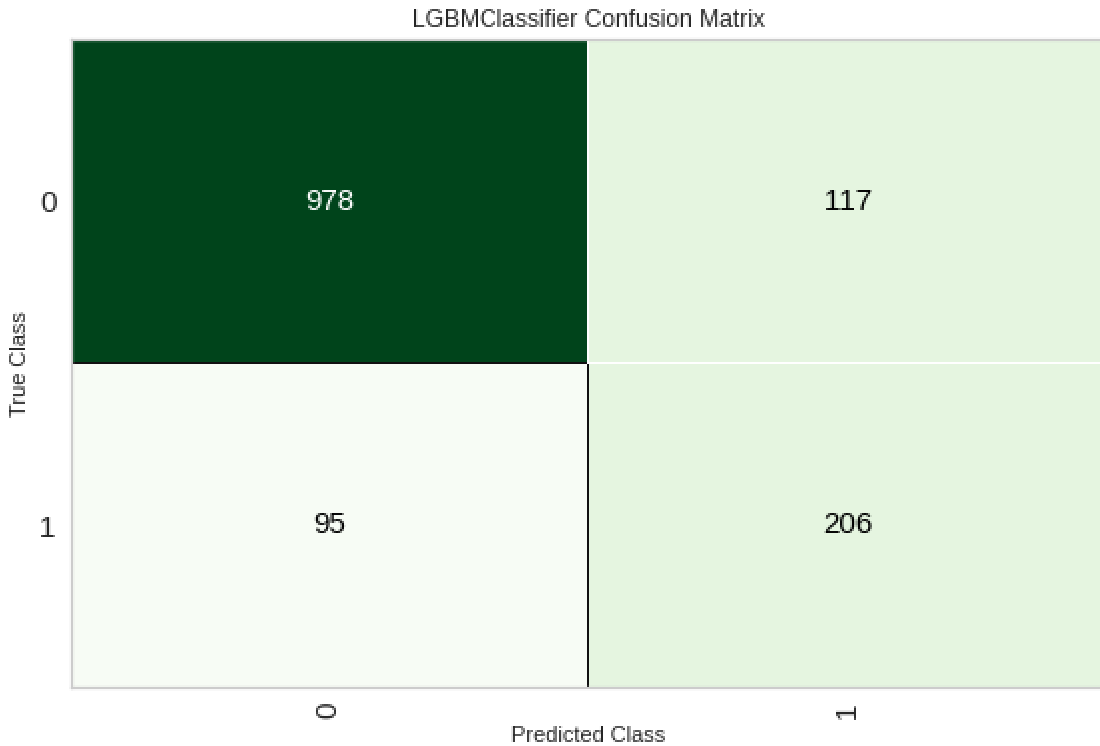

```
INFO:logs:Visual Rendered Successfully
INFO:logs:plot_model() succesfully
completed…
```

The random forest classifier is showing 46.4% precision and a mean accuracy of 76.6%. While this is slightly better than extra treees model, the performance is still not ideal. Most of the improvement comes from accurately identifying more standard wines (true negatives). Only two additional true positives were identified.

```
[ ]: # tune lgb model
tuned_lgb = tune_model(lgb, optimize='Prec.')
```

```
      Accuracy     AUC  Recall   Prec.      F1   Kappa     MCC
Fold
0       0.8436  0.8950  0.5857  0.6508  0.6165  0.5186  0.5198
1       0.8497  0.8511  0.6286  0.6567  0.6423  0.5472  0.5475
2       0.8466  0.8560  0.5857  0.6613  0.6212  0.5255  0.5270
3       0.8497  0.8589  0.5857  0.6721  0.6260  0.5325  0.5344
4       0.8650  0.8932  0.7429  0.6667  0.7027  0.6157  0.6172
5       0.8313  0.8319  0.5286  0.6271  0.5736  0.4694  0.4721
6       0.8650  0.8614  0.6620  0.7015  0.6812  0.5956  0.5961
7       0.8615  0.9282  0.8000  0.6437  0.7134  0.6235  0.6299
8       0.8462  0.8969  0.6143  0.6515  0.6324  0.5352  0.5356
9       0.8031  0.8647  0.5571  0.5417  0.5493  0.4233  0.4234
Mean    0.8462  0.8737  0.6291  0.6473  0.6359  0.5387  0.5403
Std     0.0175  0.0271  0.0802  0.0398  0.0496  0.0593  0.0601


INFO:logs:create_model_container: 20
INFO:logs:master_model_container: 20
INFO:logs:display_container: 8
INFO:logs:LGBMClassifier(bagging_fraction=0.9, bagging_freq=3,
boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.5,
               importance_type='split', learning_rate=0.4, max_depth=-1,
               min_child_samples=6, min_child_weight=0.001, min_split_gain=0.3,
               n_estimators=20, n_jobs=-1, num_leaves=150, objective=None,
               random_state=123, reg_alpha=0.005, reg_lambda=0.0005,
               silent='warn', subsample=1.0, subsample_for_bin=200000,
               subsample_freq=0)
INFO:logs:tune_model() succesfully
completed…
```

```
[ ]: plot_model(tuned_lgb, plot = 'confusion_matrix')
```




```
INFO:logs:Visual Rendered Successfully
INFO:logs:plot_model() succesfully
completed…
```

The light gradient boosting model shows the best performance on the training data by far. The classifier has 64.7% precision and 84.6% accuracy. While light gradient boosting model has the fewest true positives, it also minimized false positives (which harm the brand reputation) and false negative (which cost the client money).

### 1.7 Check Performance on Validation Set

As noted above, machine learning models tend to perform better on the training data than validation data. Before we finalize our choice of classifier, we need to test the models' performance on the validation set.

[ ]:
```
#ET performance on the validation set
pred_holdout_et = predict_model(tuned_et)
pred_holdout_et.head()
```

```
INFO:logs:Initializing predict_model()
INFO:logs:predict_model(estimator=ExtraTreesClassifier(bootstrap=False,
ccp_alpha=0.0,
                     class_weight='balanced_subsample', criterion='gini',
                     max_depth=6, max_features=1.0, max_leaf_nodes=None,
                     max_samples=None, min_impurity_decrease=0,
```

```
                     min_impurity_split=None, min_samples_leaf=4,
                     min_samples_split=7, min_weight_fraction_leaf=0.0,
                     n_estimators=200, n_jobs=-1, oob_score=False,
                     random_state=123, verbose=0, warm_start=False),
probability_threshold=None, encoded_labels=False, drift_report=False,
raw_score=False, round=4, verbose=True, ml_usecase=MLUsecase.CLASSIFICATION,
display=None, drift_kwargs=None)
INFO:logs:Checking exceptions
INFO:logs:Preloading libraries
INFO:logs:Preparing display monitor
```

```
                   Model  Accuracy     AUC  Recall   Prec.      F1   Kappa  \
0  Extra Trees Classifier    0.7278  0.8375   0.804  0.4298  0.5602  0.3883

      MCC
0  0.4282
```

[ ]:

```
   fixed acidity  volatile acidity  citric acid  chlorides  \
0      -0.236945         -0.148537     0.257667  -0.784333
1      -1.328910         -1.023989    -0.695223   0.013572
2      -0.632079         -0.419966     1.337734   0.600194
3       0.598951          1.135776    -0.905309   0.600194
4       0.818899          0.215861     0.082074   0.471080

   free sulfur dioxide  total sulfur dioxide        pH  sulphates   alcohol  \
0             0.283892             -0.370703  0.099835  -0.821698  1.652311
1            -0.024714             -0.296549  0.298438  -0.701081 -1.137028
2             1.743240              1.603736  0.363247  -1.599922 -0.803320
3            -1.096339             -0.852266 -0.536955   0.662836  0.691685
4             0.513223              0.978935  1.217709   0.957118 -0.194076

   quality  Label   Score
0        0      1  0.8592
1        0      0  0.7546
2        0      0  0.8483
3        0      0  0.5832
4        0      0  0.7107
```

[ ]:

```python
#RF performance on the validation set
pred_holdout_rf = predict_model(tuned_rf)
pred_holdout_rf.head()
```

```
INFO:logs:Initializing predict_model()
INFO:logs:predict_model(estimator=RandomForestClassifier(bootstrap=False,
ccp_alpha=0.0,
                       class_weight='balanced_subsample', criterion='gini',
                       max_depth=6, max_features='log2', max_leaf_nodes=None,
                       max_samples=None, min_impurity_decrease=0.001,
```

```
                          min_impurity_split=None, min_samples_leaf=6,
                          min_samples_split=9, min_weight_fraction_leaf=0.0,
                          n_estimators=190, n_jobs=-1, oob_score=False,
                          random_state=123, verbose=0, warm_start=False),
probability_threshold=None, encoded_labels=False, drift_report=False,
raw_score=False, round=4, verbose=True, ml_usecase=MLUsecase.CLASSIFICATION,
display=None, drift_kwargs=None)
INFO:logs:Checking exceptions
INFO:logs:Preloading libraries
INFO:logs:Preparing display monitor
```

```
                      Model  Accuracy     AUC  Recall   Prec.      F1   Kappa  \
0  Random Forest Classifier    0.7514  0.8463  0.8106  0.4569  0.5844  0.4262

      MCC
0  0.4618
```

[ ]:

```
   fixed acidity  volatile acidity  citric acid  chlorides  \
0      -0.236945         -0.148537     0.257667  -0.784333
1      -1.328910         -1.023989    -0.695223   0.013572
2      -0.632079         -0.419966     1.337734   0.600194
3       0.598951          1.135776    -0.905309   0.600194
4       0.818899          0.215861     0.082074   0.471080

   free sulfur dioxide  total sulfur dioxide        pH  sulphates   alcohol  \
0             0.283892             -0.370703  0.099835  -0.821698  1.652311
1            -0.024714             -0.296549  0.298438  -0.701081 -1.137028
2             1.743240              1.603736  0.363247  -1.599922 -0.803320
3            -1.096339             -0.852266 -0.536955   0.662836  0.691685
4             0.513223              0.978935  1.217709   0.957118 -0.194076

   quality  Label   Score
0        0      1  0.8370
1        0      0  0.8497
2        0      0  0.9001
3        0      0  0.7342
4        0      0  0.7873
```

[ ]:

```
#LGB performance on the validation set
pred_holdout_lgb = predict_model(tuned_lgb)
pred_holdout_lgb.head()
```

```
INFO:logs:Initializing predict_model()
INFO:logs:predict_model(estimator=LGBMClassifier(bagging_fraction=0.9,
bagging_freq=3, boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.5,
               importance_type='split', learning_rate=0.4, max_depth=-1,
               min_child_samples=6, min_child_weight=0.001, min_split_gain=0.3,
```

```
               n_estimators=20, n_jobs=-1, num_leaves=150, objective=None,
               random_state=123, reg_alpha=0.005, reg_lambda=0.0005,
               silent='warn', subsample=1.0, subsample_for_bin=200000,
               subsample_freq=0), probability_threshold=None,
encoded_labels=False, drift_report=False, raw_score=False, round=4,
verbose=True, ml_usecase=MLUsecase.CLASSIFICATION, display=None,
drift_kwargs=None)
INFO:logs:Checking exceptions
INFO:logs:Preloading libraries
INFO:logs:Preparing display monitor

                             Model  Accuracy     AUC  Recall   Prec.      F1  \
0  Light Gradient Boosting Machine    0.8481  0.8873  0.6844  0.6378  0.6603

    Kappa     MCC
0  0.5626  0.5632
```

[ ]:
```
   fixed acidity  volatile acidity  citric acid  chlorides  \
0      -0.236945         -0.148537     0.257667  -0.784333
1      -1.328910         -1.023989    -0.695223   0.013572
2      -0.632079         -0.419966     1.337734   0.600194
3       0.598951          1.135776    -0.905309   0.600194
4       0.818899          0.215861     0.082074   0.471080

   free sulfur dioxide  total sulfur dioxide        pH  sulphates   alcohol  \
0             0.283892             -0.370703  0.099835  -0.821698  1.652311
1            -0.024714             -0.296549  0.298438  -0.701081 -1.137028
2             1.743240              1.603736  0.363247  -1.599922 -0.803320
3            -1.096339             -0.852266 -0.536955   0.662836  0.691685
4             0.513223              0.978935  1.217709   0.957118 -0.194076

   quality  Label   Score
0        0      1  0.9891
1        0      0  0.9937
2        0      0  0.9987
3        0      0  0.9486
4        0      0  0.9569
```

The light gradient boosting model still has the best precision and accuracy out of our three classifiers. Moreover, the performance on the validation set is not much lower than the training performance. This indicates that our model is not overfit and can generalise relatively well to unseen data. Therefore, we will finalize the light gradient boosting model to present to the client.

[ ]:
```
#Finalize model and retrain on the combined training and validation sets before
 ↪testing on the 5% holdout
final_lgb = finalize_model(tuned_lgb)
```

```
INFO:logs:Initializing finalize_model()
```

```
INFO:logs:finalize_model(estimator=LGBMClassifier(bagging_fraction=0.9,
bagging_freq=3, boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.5,
               importance_type='split', learning_rate=0.4, max_depth=-1,
               min_child_samples=6, min_child_weight=0.001, min_split_gain=0.3,
               n_estimators=20, n_jobs=-1, num_leaves=150, objective=None,
               random_state=123, reg_alpha=0.005, reg_lambda=0.0005,
               silent='warn', subsample=1.0, subsample_for_bin=200000,
               subsample_freq=0), fit_kwargs=None, groups=None, model_only=True,
display=None, experiment_custom_tags=None, return_train_score=False)
INFO:logs:Finalizing LGBMClassifier(bagging_fraction=0.9, bagging_freq=3,
boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.5,
               importance_type='split', learning_rate=0.4, max_depth=-1,
               min_child_samples=6, min_child_weight=0.001, min_split_gain=0.3,
               n_estimators=20, n_jobs=-1, num_leaves=150, objective=None,
               random_state=123, reg_alpha=0.005, reg_lambda=0.0005,
               silent='warn', subsample=1.0, subsample_for_bin=200000,
               subsample_freq=0)
INFO:logs:Initializing create_model()
INFO:logs:create_model(estimator=LGBMClassifier(bagging_fraction=0.9,
bagging_freq=3, boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.5,
               importance_type='split', learning_rate=0.4, max_depth=-1,
               min_child_samples=6, min_child_weight=0.001, min_split_gain=0.3,
               n_estimators=20, n_jobs=-1, num_leaves=150, objective=None,
               random_state=123, reg_alpha=0.005, reg_lambda=0.0005,
               silent='warn', subsample=1.0, subsample_for_bin=200000,
               subsample_freq=0), fold=None, round=4, cross_validation=True,
predict=True, fit_kwargs={}, groups=None, refit=True, verbose=False,
system=False, metrics=None, experiment_custom_tags=None,
add_to_model_list=False, probability_threshold=None, display=None,
return_train_score=False, kwargs={})
INFO:logs:Checking exceptions
INFO:logs:Importing libraries
INFO:logs:Copying training dataset
INFO:logs:Defining folds
INFO:logs:Declaring metric variables
INFO:logs:Importing untrained model
INFO:logs:Declaring custom model
INFO:logs:Light Gradient Boosting Machine Imported succesfully
INFO:logs:Starting cross validation
INFO:logs:Cross validating with StratifiedKFold(n_splits=10, random_state=None,
shuffle=False), n_jobs=-1
INFO:logs:Calculating mean and std
INFO:logs:Creating metrics dataframe
INFO:logs:Finalizing model
INFO:logs:create_model_container: 20
```

```
INFO:logs:master_model_container: 20
INFO:logs:display_container: 12
INFO:logs:LGBMClassifier(bagging_fraction=0.9, bagging_freq=3,
boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.5,
               importance_type='split', learning_rate=0.4, max_depth=-1,
               min_child_samples=6, min_child_weight=0.001, min_split_gain=0.3,
               n_estimators=20, n_jobs=-1, num_leaves=150, objective=None,
               random_state=123, reg_alpha=0.005, reg_lambda=0.0005,
               silent='warn', subsample=1.0, subsample_for_bin=200000,
               subsample_freq=0)
INFO:logs:create_model() succesfully
completed…
INFO:logs:create_model_container: 20
INFO:logs:master_model_container: 20
INFO:logs:display_container: 11
INFO:logs:LGBMClassifier(bagging_fraction=0.9, bagging_freq=3,
boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.5,
               importance_type='split', learning_rate=0.4, max_depth=-1,
               min_child_samples=6, min_child_weight=0.001, min_split_gain=0.3,
               n_estimators=20, n_jobs=-1, num_leaves=150, objective=None,
               random_state=123, reg_alpha=0.005, reg_lambda=0.0005,
               silent='warn', subsample=1.0, subsample_for_bin=200000,
               subsample_freq=0)
INFO:logs:finalize_model() succesfully
completed…
```

[ ]:
```
print(final_lgb)
```

```
LGBMClassifier(bagging_fraction=0.9, bagging_freq=3, boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.5,
               importance_type='split', learning_rate=0.4, max_depth=-1,
               min_child_samples=6, min_child_weight=0.001, min_split_gain=0.3,
               n_estimators=20, n_jobs=-1, num_leaves=150, objective=None,
               random_state=123, reg_alpha=0.005, reg_lambda=0.0005,
               silent='warn', subsample=1.0, subsample_for_bin=200000,
               subsample_freq=0)
```

### 1.8 Predict on Unseen Data

As a demonstration of how the model works, we predict the quality of wines in the 5% holdout data that we removed at the begining of the notebook before feature engineering.

[ ]:
```
# 5% sample witheld in the beginning
data_unseen.head()
```

```
[ ]:    fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
     0            8.1              0.28         0.40             6.9      0.050
     1            8.6              0.23         0.40             4.2      0.035
     2            6.6              0.16         0.40             1.5      0.044
     3            7.4              0.34         0.42             1.1      0.033
     4            6.0              0.19         0.26            12.4      0.048

        free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
     0                 30.0                  97.0   0.9951  3.26       0.44
     1                 17.0                 109.0   0.9947  3.14       0.53
     2                 48.0                 143.0   0.9912  3.54       0.52
     3                 17.0                 171.0   0.9917  3.12       0.53
     4                 50.0                 147.0   0.9972  3.30       0.36

        alcohol  quality
     0     10.1        0
     1      9.7        0
     2     12.4        1
     3     11.3        0
     4      8.9        0
```

```
[ ]: # drop the quality column (classification label) from data_unseen
     data_unseen.drop('quality', axis = 1, inplace = True)
     data_unseen.head()
```

```
[ ]:    fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
     0            8.1              0.28         0.40             6.9      0.050
     1            8.6              0.23         0.40             4.2      0.035
     2            6.6              0.16         0.40             1.5      0.044
     3            7.4              0.34         0.42             1.1      0.033
     4            6.0              0.19         0.26            12.4      0.048

        free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
     0                 30.0                  97.0   0.9951  3.26       0.44
     1                 17.0                 109.0   0.9947  3.14       0.53
     2                 48.0                 143.0   0.9912  3.54       0.52
     3                 17.0                 171.0   0.9917  3.12       0.53
     4                 50.0                 147.0   0.9972  3.30       0.36

        alcohol
     0     10.1
     1      9.7
     2     12.4
     3     11.3
     4      8.9
```

[ ]:
```
#predict class using the finalized model
pred_unseen = predict_model(final_lgb, data=data_unseen)
pred_unseen.head()
```

```
INFO:logs:Initializing predict_model()
INFO:logs:predict_model(estimator=LGBMClassifier(bagging_fraction=0.9,
bagging_freq=3, boosting_type='gbdt',
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.5,
               importance_type='split', learning_rate=0.4, max_depth=-1,
               min_child_samples=6, min_child_weight=0.001, min_split_gain=0.3,
               n_estimators=20, n_jobs=-1, num_leaves=150, objective=None,
               random_state=123, reg_alpha=0.005, reg_lambda=0.0005,
               silent='warn', subsample=1.0, subsample_for_bin=200000,
               subsample_freq=0), probability_threshold=None,
encoded_labels=False, drift_report=False, raw_score=False, round=4,
verbose=True, ml_usecase=MLUsecase.CLASSIFICATION, display=None,
drift_kwargs=None)
INFO:logs:Checking exceptions
INFO:logs:Preloading libraries
INFO:logs:Preparing display monitor
```

[ ]:
```
   fixed acidity  volatile acidity  citric acid  residual sugar  chlorides  \
0            8.1              0.28         0.40             6.9      0.050
1            8.6              0.23         0.40             4.2      0.035
2            6.6              0.16         0.40             1.5      0.044
3            7.4              0.34         0.42             1.1      0.033
4            6.0              0.19         0.26            12.4      0.048

   free sulfur dioxide  total sulfur dioxide  density    pH  sulphates  \
0                 30.0                  97.0   0.9951  3.26       0.44
1                 17.0                 109.0   0.9947  3.14       0.53
2                 48.0                 143.0   0.9912  3.54       0.52
3                 17.0                 171.0   0.9917  3.12       0.53
4                 50.0                 147.0   0.9972  3.30       0.36

   alcohol  Label   Score
0     10.1      0  0.9946
1      9.7      0  0.9884
2     12.4      1  0.8064
3     11.3      0  0.6941
4      8.9      0  0.9858
```

[ ]:
```
#save model
save_model(final_lgb,'lgb_final_pipeline')
```

```
INFO:logs:Initializing save_model()
INFO:logs:save_model(model=LGBMClassifier(bagging_fraction=0.9, bagging_freq=3,
boosting_type='gbdt',
```

```
               class_weight=None, colsample_bytree=1.0, feature_fraction=0.5,
               importance_type='split', learning_rate=0.4, max_depth=-1,
               min_child_samples=6, min_child_weight=0.001, min_split_gain=0.3,
               n_estimators=20, n_jobs=-1, num_leaves=150, objective=None,
               random_state=123, reg_alpha=0.005, reg_lambda=0.0005,
               silent='warn', subsample=1.0, subsample_for_bin=200000,
               subsample_freq=0), model_name=lgb_final_pipeline,
prep_pipe_=Pipeline(memory=None,
         steps=[('dtypes',
                 DataTypes_Auto_infer(categorical_features=[],
                                      display_types=True, features_todrop=[],
                                      id_columns=[],
                                      ml_usecase='classification',
                                      numerical_features=[], target='quality',
                                      time_features=[])),
                ('imputer',
                 Simple_Imputer(categorical_strategy='not_available',
                                fill_value_categorical=None,
                                fill_value_numerical=None,
                                numeric_stra…
                ('dummy', Dummify(target='quality')),
                ('fix_perfect', Remove_100(target='quality')),
                ('clean_names', Clean_Colum_Names()),
                ('feature_select', 'passthrough'),
                ('fix_multi',
                 Fix_multicollinearity(correlation_with_target_preference=None,
                                       correlation_with_target_threshold=0.0,
                                       target_variable='quality',
                                       threshold=0.7)),
                ('dfs', 'passthrough'), ('pca', 'passthrough')],
         verbose=False), verbose=True, kwargs={})
INFO:logs:Adding model into prep_pipe
INFO:logs:lgb_final_pipeline.pkl saved in current working directory
INFO:logs:Pipeline(memory=None,
         steps=[('dtypes',
                 DataTypes_Auto_infer(categorical_features=[],
                                      display_types=True, features_todrop=[],
                                      id_columns=[],
                                      ml_usecase='classification',
                                      numerical_features=[], target='quality',
                                      time_features=[])),
                ('imputer',
                 Simple_Imputer(categorical_strategy='not_available',
                                fill_value_categorical=None,
                                fill_value_numerical=None,
                                numeric_stra…
                                colsample_bytree=1.0, feature_fraction=0.5,
                                importance_type='split', learning_rate=0.4,
```

```
                                    max_depth=-1, min_child_samples=6,
                                    min_child_weight=0.001, min_split_gain=0.3,
                                    n_estimators=20, n_jobs=-1, num_leaves=150,
                                    objective=None, random_state=123,
                                    reg_alpha=0.005, reg_lambda=0.0005,
                                    silent='warn', subsample=1.0,
                                    subsample_for_bin=200000, subsample_freq=0)]],
          verbose=False)
INFO:logs:save_model() successfully
completed…

Transformation Pipeline and Model Successfully Saved
```

```
[ ]: (Pipeline(memory=None,
               steps=[('dtypes',
                       DataTypes_Auto_infer(categorical_features=[],
                                            display_types=True, features_todrop=[],
                                            id_columns=[],
                                            ml_usecase='classification',
                                            numerical_features=[], target='quality',
                                            time_features=[])),
                      ('imputer',
                       Simple_Imputer(categorical_strategy='not_available',
                                      fill_value_categorical=None,
                                      fill_value_numerical=None,
                                      numeric_stra…
                                      colsample_bytree=1.0, feature_fraction=0.5,
                                      importance_type='split', learning_rate=0.4,
                                      max_depth=-1, min_child_samples=6,
                                      min_child_weight=0.001, min_split_gain=0.3,
                                      n_estimators=20, n_jobs=-1, num_leaves=150,
                                      objective=None, random_state=123,
                                      reg_alpha=0.005, reg_lambda=0.0005,
                                      silent='warn', subsample=1.0,
                                      subsample_for_bin=200000, subsample_freq=0)]],
               verbose=False), 'lgb_final_pipeline.pkl')
```

### 1.9 Conclusions

Therefore, we selected the light gradient boosting model for this problem as it had the best precision and accuracy. This model will help our client LCBO make a better decision where assessing white wine quality and will be in better position to price white whites in their stores to get the best balance of maximizing profit while maintaining a good reputation.

**Link to App**: https://wineplus.herokuapp.com/

## 1.10 References

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Mani, S., Krishnankutty, R. A., Swaminathan, S., & Theerthagiri, P. (2023). An investigation of wine quality testing using machine learning techniques. IAES International Journal of Artificial Intelligence, 12(2), 747.